On the Enhancement of Intelligibility

Investigating the influence of different speech modifications on the intelligibility of speech in near-end noise

by

Bob Luppes
Ellen Riemens

21-06-2019

For an Automatic Volume Control System
in context of the
Bachelor Graduation Project

Thesis committee:	prof. dr. P.M. Sarro,	TU Delft, Chair
	Dr. ir. R.C. Hendriks,	TU Delft, Supervisor
	Dr. J. Martinez,	TU Delft, Jury member
	Dr. ir. A. Koutrouvelis,	TU Delft, Daily supervisor

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Intentionally left blank

Abstract

Several algorithms to enhance the intelligibility of speech in near-end noise were analyzed and implemented. The algorithms considered were assessed based on the intrusive instrumental intelligibility metric SIIB^{Gauss}. An implementation based on the direct optimization for this metric is assessed, as well as an implementation based on human induced speech modifications, including increased sound intensity, flattening of the spectral tilt, increased vowel duration and increased consonant-vowel ratio. Another implemented algorithm is the amplification of the transient component of speech. Results show that for increased vowel duration a decrease in intelligibility was found in SIIB^{Gauss} value as well as in informal listening tests. The other implementations did show an increase in intelligibility according to SIIB^{Gauss} at SNRs between -4 dB and 6 dB in both stationary and fluctuating noise, under a power constraint. Finally, the implementations were combined into a system that automatically selects the optimal algorithm to use under the given noise conditions. It is shown that this combined system is able to increase intelligibility of speech in the presence of non-fluctuating noise, fluctuating noise, speech shaped noise, and competing speaker noise.

Preface

This thesis is written by order of the Bachelor Graduation Project of Electrical Engineering. We would like to thank John Schmitz, who proposed the idea for this thesis. We would also like to express our gratitude to our supervisor dr. ir. Richard Hendriks and postdoc dr. ir. Andreas Koutrouvelis, who were always available for questions.

We would also like to thank researchers in the area of intelligibility enhancement for making their material available. In particular we would like to thank C. Taal and S. Van Kuyk for making their Matlab implementations available on the public domain.

Lastly, we would like our colleagues: Thijs Timmer, Quinten van Wingerden, Mats Rijkeboer, and Winay Sewnarain for all our good discussions and a pleasant collaboration.

Bob Luppes
Ellen Riemens
Delft, June 2019

Contents

1 Introduction

1.1. Problem Definition

In noisy environments it can often be difficult to understand speech played over a speaker system. Take for example the situation where a pre-recorded message is broadcast over a public address system (PA system) at a station while a train is passing by. Or, for instance, the situation where speech is played inside a car in the presence of noise from the engine and tires. Apart from these, there are numerous other situations in which the intelligibility of speech is degraded due to the presence of noise in the room containing the speaker system (near-end noise).

The nature of this near-end noise may vary, depending on the situation. A difference is made between fluctuating and non-fluctuating (wide sense stationary) noise. These two noise conditions may arise at a train station, for instance, when multiple trains are passing by and the train noise is considered fluctuating, or background chatter in a station, which can be assumed to be wide sense stationary. Also, a difference is made between different signal to noise ratios (SNRs), which is a measure of how strong the speech signal power is compared to the noise power.

A system is developed in order to solve this problem. There are two approaches to make speech more intelligible in the presence of near-end noise. First, in small spaces, such as a car, it is possible to suppress the near-end noise and thereby improve the intelligibility of the speech signal.

Also, in contrary to music, the important part of speech is the message that is contained within the speech signal and not the actual signal itself. Therefore, it is possible to use signal processing in order to alter the speech signal. This introduces a distortion but at the same time it is possible to increase the intelligibility of the speech signal in the presence of near-end noise.

The system is divided into three subsystems, namely the *noise statistics estimation subgroup*, the *intelligibility enhancement subgroup* and the *amplifier and noise cancellation subgroup*.

The *noise statistics estimation subgroup* estimates the near-end noise, which is used by the *intelligibility enhancement subgroup*. This is done by processing audio recordings of the environment in which the speaker system is deployed.

The *intelligibility enhancement subgroup* uses this information about the near-end noise to alter the input speech signal in a way such that it will be more intelligible in the presence of the near-end noise. This enhanced speech signal is used by the *amplifier and noise cancellation subgroup* as well as the *noise statistics estimation subgroup*. Apart from this, also the amplification factor that is needed to achieve a predefined word recognition rate of the enhanced speech signal is calculated and sent to the *amplifier and noise cancellation subgroup*.

In its turn, the amplifier as build by the *amplifier and noise cancellation subgroup* is then used to play the amplified enhanced speech signal on an existing speaker system. The amplifier is also capable of active noise cancellation, for which it uses an extra microphone placed in the noisy environment. The subdivision as described above can be seen in figure 1.1.

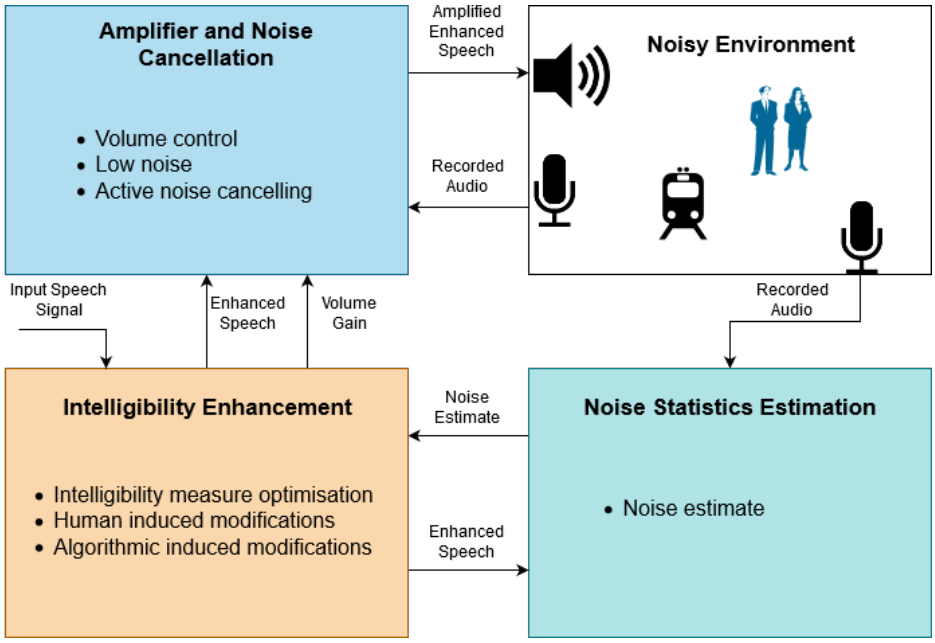


Figure 1.1: System overview.

In this thesis, the design of the *intelligibility enhancement subsystem* will be discussed. This includes the three implementations as depicted the intelligibility enhancement block in figure 1.1; optimization for an intelligibility measure, human speech modifications, and algorithmic speech modifications. How these algorithms work and how they should be implemented is the main focus of this thesis.

This subsystem will be designed alongside the other two subsystems and will eventually be combined in order to make the full system as depicted in figure 1.1. From here on out, this full system will be referenced to as the overall system.

1.2. State of the Art Analysis

As described in the problem definition, section 1.1, real-time speech or pre-recorded speech is often played in the presence of near-end noise. For a listener, this will result in a reduced intelligibility of the speech signal. A clear distinction is made between algorithms that enhance the intelligibility of speech signals affected by noise at the microphone side (far-end noise) and algorithms that process clean speech signals to improve the intelligibility in the presence of noise on the speaker system side (near-end noise). In this thesis, we will focus on the latter.

In order to effectively assess the intelligibility of the speech signal, it is necessary to have an objective measure that is easily calculated, as to avoid expensive and time consuming listening tests. These measures take into account the workings of the human ear and are accurate measures for intelligibility under certain circumstances. These circumstances are different for each measure. Many such measures exist and are classified as intrusive instrumental intelligibility metrics, when the clean speech and the distorted speech signal are necessary to assess intelligibility [1]. Examples of such metrics are the Speech Intelligibility Index (SII) [2], which is a metric based on the long-term average SNR per frequency band, the Short-Time Objective Intelligibility (STOI) [3], which evaluates the correlation coefficient between clean and noisy speech for short time-frames , the Speech Intelligibility In Bits (SIIB) [4], which is a measure for the mutual information between the clean speech and noisy speech , and SIIB^{Gauss} [1], which is SIIB under the assumption of a Gaussian channel.

From here on out, these intrusive instrumental intelligibility metrics will be referenced to as objective intelligibility measures.

To increase intelligibility, a straightforward approach would be to directly optimize for one of these measures. Taal et al. has implemented and investigated the algebraic optimization of SII [5] and concluded that

optimizing for SII highly correlates with the word recognition rate in listening tests. Goli et al.[6] did a similar optimization for STOI, in which it was concluded by listening tests that the word recognition rate was improved.

Different algorithms have been proposed to enhance clean speech signals such that they are more intelligible in the presence of near-end noise without directly optimizing for a specific intelligibility measure. Cooke et al. [7] investigated the effect of different speech modifications on the intelligibility under noisy conditions. Based on their work, a distinction can be made between human induced speech modifications and algorithmic induced speech modifications, of which the latter includes, amongst other things, direct optimization for an intelligibility measure.

Two natural speech effects as exhibited by humans can be distinguished; Lombard speech [8][9]. and clear speech [10]. These effects are involuntary and arise when the situation requires that speech is more intelligible for the listener.

Lombard speech occurs when humans talk in noisy environments and it incorporates multiple speech modifications compared to normal speech. These modifications include an increased sound intensity, an increase in phonetic fundamental frequency, a flattening of the spectral tilt, an increased vowel duration, and a shift in formant center frequencies. For many of these modifications it has been tried to replicate replicate them using algorithms, which are referred to as Lombard algorithms. Jokinen et al. [11] developed an adaptive post-filtering method for intelligibility enhancement of narrowband telephone speech producing an artificial Lombard-like effect.

Another way humans alter their speech is clear speech. This occurs when speakers are told to speak 'clearly' or when talking to hearing-impaired listeners [10]. Speech modifications of clear speech include a flattening of the spectral tilt, an increased duration, and an increased consonant-vowel ratio.

Algorithmic induced speech modifications are for instance the amplification of the transient component of speech, dynamic range compression and spectro-temporal energy reallocation by optimization for an objective intelligibility measure. The latter is already discussed above.

In Yoo et al. [12], it is shown that the transient component of speech contributes greatly to the intelligibility. Several algorithms have been developed to extract this transient component. Yoo et al. [13] developed an algorithm based on time-varying band-pass filters to remove formant energy from the speech signal, the residual is considered to be mostly transients. Tantibundhit et al. [14] considered another approach in which tonal component of speech are extracted using MDCT-based hidden markov chain models and transient componants are extracted using wavelet-based hidden markov tree models. In 2011, Rasetshwane et al. [15] proposed two simplified approaches of these techniques, showing similar results but having less computational complexity. One is based on the wavelet-packet transform as proposed by Tantibundhit, while the other implements a static filter based on the time-varying band-pass filers by Yoo.

Dynamic range compression is often used to reduce the amplitude of loud sounds and increase the amplitude of quiet sounds, compressing the range of amplitudes in the signal. In Niederjohn et al. [16], high-pass filtering is combined with dynamic range compression and the resulting intelligibility is compared to the intelligibility of a speech signal that is only high-pass filtered, where a substantial improvement was found. In Zorila et al.[17], dynamic range compression is combined with spectral tilting to improve intelligibility of speech in noise, finding an improvement in intelligibility in speech shaped noise as well as competing speaker noise.

1.3. Chapter Organization

In the following chapters, the design and implementation of the *intelligibility enhancement subsystem* will be discussed and analyzed.

Chapter 2 describes the programme of requirements for the overall system as shown in figure 1.1, while chapter 3 describes the design criteria for the subsystem to be designed. Chapter 4 briefly explains some necessary theory for the algorithms discussed in the following chapters. Chapter 5 provides an overview of the existing speech modification techniques and gives an analysis of these. The implementation of the suitable techniques based on this analysis are described in chapter 6. Based on the preliminary results of the implemented speech modifications, these algorithms are combined into the full *intelligibility enhancement subsystem* in chapter 7. The result in terms of increased intelligibility that is achieved with this subsystem is presented in chapter 8 while the implications of these results are discussed and recommendations for future work are given in chapter 9.

2

Programme of Requirements

The full system in figure 1.1 needs to solve the intelligibility problem as described in section 1.1. In order to solve this problem, the full system will have to satisfy certain requirements, which are described in this chapter. Later, they will be used as a starting point for the design choices of the *intelligibility enhancement subsystem* and the other two subsystems.

In specifying these requirements for the overall system, some assumptions were made regarding the noise conditions and the existing hardware in the near-end environment. These assumptions can be found below.

Assumptions

1. The near-end noise is uncorrelated with the speech signal.
    - This is a reasonable assumption when near-end noise is defined as a signal that consists of contributions of all noise sources except the loudspeaker.
2. The input of the existing power amplifier needs an audio input at a standardized line level of 0.447 V_{max} [18].
    - Typical value for consumer applications.
3. The gain of the existing power amplifier is equal to 25.
    - A set gain of the existing system is needed, to determine how much the overall system needs to amplify the signal in order to reach a certain output level at the speaker.
4. The output level of the existing power amplifier is less than 100 dBA.
    - From [19], the maximum permissible occupational noise exposure for 2 hours is 100 dBA. Assuming that PA system employees work for 8 hours per day, this means the PA system can be used for announcements 25% of the time.
5. The input signal is pre-recorded and noise-free.
    - Typical for announcements, music and audio-books.

Several requirements need to be defined in order to make a successful design. These requirements are used in choosing the design criteria for the *intelligibility enhancement subsystem*.

Mandatory Requirements

The mandatory requirements need to be met in order for the overall system to be considered successful. A subdivision is made between functional and non-functional requirements, which represent requirements describing what the system has to do and requirements describing the quality of the system respectively.

1. **Functional Requirements**
   (a) The system must improve intelligibility such that the word recognition rate is at least 90 % in the presence of near-end noise.
   (b) The system must be able to suppress near-end noise in the frequency band from 0 Hz to 500 Hz.
   (c) The system must be able to process speech in the frequency band from 0 Hz to 8 kHz.
   (d) The system must be able to play audio in the frequency band from 0 Hz to 20 kHz.
   (e) The system must be able to process pre-recorded speech in advance (pre-processing).
2. **Non-Functional Requirements**
   (a) The system must operate in SNRs below 15 dB.
   (b) The system must not damage hearing.
   (c) The system must not add more than 3dBA noise to the enhanced speech signal.
   (d) The system must have a maximum pre-processing delay of 5 times the duration of the input signal with a maximum of 20 minutes.
   (e) The system must have a maximum latency of 100 ms without pre-processing.
   (f) The system must not record near-end noise when the system is not broadcasting any audio.
   (g) The system must not store recorded near-end noise longer than the system latency.

Trade-Off Requirements

The trade-off requirements are not necessary to be met, however, if the they are met, the end-users become increasingly satisfied.

1. The system should be plug-and-play, using a microcontroller.
2. The system should work on a 12 V-input.
3. The system should not add more than 1 dBA noise to the enhanced speech signal.
4. The sound coming out of the system should sound natural according to listening tests.
5. The combined price of all the individual components should not exceed 100 euro.

From here on out, these requirements will be referenced to as overall requirements.

3

Design Criteria for the Intelligibility Enhancement Subsystem

Based on the programme of requirements in chapter 2, the design criteria can be specified for the *intelligibility enhancement subsystem*, needed to satisfy the mandatory and trade-off requirements. The input and output specifications are given in section 3.1, these define the interface between the *intelligibility enhancement subsystem* and the other subsystems. After that, the requirements of the subsystem, based on the overall requirements, are given in section 3.2.

Not all of the overall requirements are relevant for the *intelligibility enhancement subsystem*, since some are influenced solely by other subsystems. The requirements influenced by this subsystem are listed below.

Relevant Overall Requirements

- **Mandatory Requirements**
  - **Overall requirement 1a**: The system must improve intelligibility such that the word recognition rate is at least 90 % in the presence of near-end noise.
  - **Overall requirement 1c**: The system must be able to process speech in the frequency band from 0 to 8 kHz.
  - **Overall requirement 1e**: The system must be able to process pre-recorded speech in advance (pre-processing).
  - **Overall requirement 2d**: The system must have a maximum pre-processing delay of 5 times the duration of the input signal with a maximum of 20 minutes.
  - **Overall requirement 2e**: The system must have a maximal latency of 100 ms with pre-processing.
  - **Overall requirement 2g**: The system must not store recorded audio longer than the system latency.
- **Trade-Off Requirements**
  - **Overall requirement 1**: The system should be plug-and-play, using a microcontroller.
  - **Overall requirement 2**: The system should work on a 12 V-input.
  - **Overall requirement 4**: The sound coming out of the product should sound natural according to listening tests.
  - **Overall requirement 5**: The combined price of all the individual components should not exceed 100 euro.

From here on out, these requirements will be referenced to as relevant overall requirements.

Based on these relevant overall requirements, key performance indicators are specified that can be used to assess the quality of the *intelligibility enhancement subsystem*.

Key Performance Indicators

The performance of the subsystem can be evaluated on different aspects. First, there is the word recognition rate, as used in listening tests, that can be used to assess the intelligibility of a clean speech signal in a noisy environment (relevant overall requirement 1a).

The latency of this subsystem directly influences the overall latency of the overall system (relevant overall requirement 2e). Together with the latency of the other two subsystems, this should not exceed 100 ms.

Since this implementation is a prototype, it can eventually be realized on a micro-controller. In order to evaluate the feasibility of this realization, the time and space complexities of this subsystem are key performance indicators as well (relevant overall requirements 2d, 1, 2 and 5).

1. Word recognition rate
2. Latency
3. Time complexity
4. Space complexity

3.1. Input and Output Specification

In figure 1.1, the input and output signals of the intelligibility subsystem can be found. These signals fully describe the interface between this subsystem and the other two subsystems. For clarity, a version of this figure is repeated in figure 3.1 including the terminology of the relevant signals as used in this thesis.

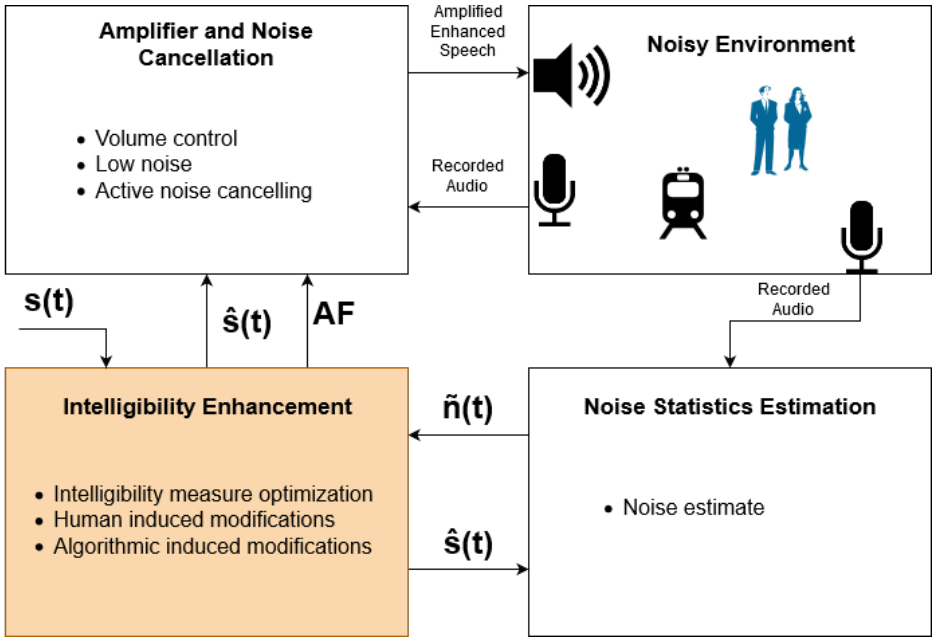


Figure 3.1: System overview.

Input Specification

The *intelligibility enhancement subsystem* requires two input signals; the clean speech signal $s(t)$ and the noise estimate $\tilde{n}(t)$. The clean speech signal is the signal to be optimized and is therefore a required input. The noise estimate, as calculated by the *noise statistics estimation subgroup* (see figure 3.1), is used for the objective intelligibility measure optimization and in order to assess the intelligibility of the enhanced speech signal $\hat{s}(t)$.

- Clean speech signal $s(t)$
  - Digital signal
  - 8000 Hz $\leq F_s \leq$ 16000 Hz
  - Noise free
- Noise estimate $\tilde{n}(t)$
  - Digital signal
  - $F_s \geq$ 16000 Hz
  - Estimated during the previous 20 ms

Output Specification

The outputs of the system are the enhanced speech signal $\hat{s}(t)$, which will be more intelligible in the presence of near-end noise, and an amplification factor, which is used by the *amplifier and noise cancellation subgroup* to amplify the enhanced signal in order to achieve the target word recognition rate as per relevant overall requirement 1a.

- Enhanced speech signal $\hat{s}(t)$
  - Digital signal
  - 8000 Hz $\leq F_s \leq$ 16000 Hz
- Amplification factor AF
  - Value between 0 and 100, with 1 being an amplification of 0.1, 50 an amplification of 1 and 100 an amplification of 10. The values in between are mapped linearly.
  - 32 bits floating-point
  - Serial signal

3.2. Requirements

Just like the programme of requirements in chapter 2, the requirements of the *intelligibility enhancement subsystem* are divided into mandatory and trade-off requirements. The mandatory requirements need to be met in order for the design to be considered successful. If the trade-off requirements are met, the end-users become increasingly satisfied.

Mandatory Requirements

1. The system must improve intelligibility such that the word recognition rate is at least 90 % in the presence of near-end noise.
2. The algorithms used in the intelligibility subsystem must be compatible with signals band-limited up to 8000 Hz.
3. The intelligibility should be maximized using energy redistribution under a power constraint, before applying a gain to reach at least 90 % word recognition.
4. The system must have a maximum pre-processing delay of 5 times the duration of the input signal with a maximum of 20 minutes.
5. The intelligibility subsystem must have a maximum latency of 30 ms.
6. The estimated near-end noise as received by the *noise statistics estimation group* must be deleted after processing.

Trade-off requirements

1. The size of the implemented program should not exceed 248 KB.
2. The space complexity of the implemented program should not exceed 8 KB.
3. The enhanced audio as computed by the intelligibility subsystem should sound natural according to listening tests.

From here on out, these requirements will simply be referenced to as mandatory and trade-off requirements.

4

Speech Formants

This chapter provides a brief description of speech formants, since this information is necessary to understand the algorithms in the chapter 5 and chapter 6.

Speech formants are frequency bands that contain relatively high power when speaking, resulting from the acoustic resonance of the human vocal tract. Speech signals can be time-frequency decomposed, which enables speech formants to be identified per time-frame. The time-frequency decomposition of a sample speech signal in which three Dutch vowels are spoken out loud can be seen in figure 4.1.

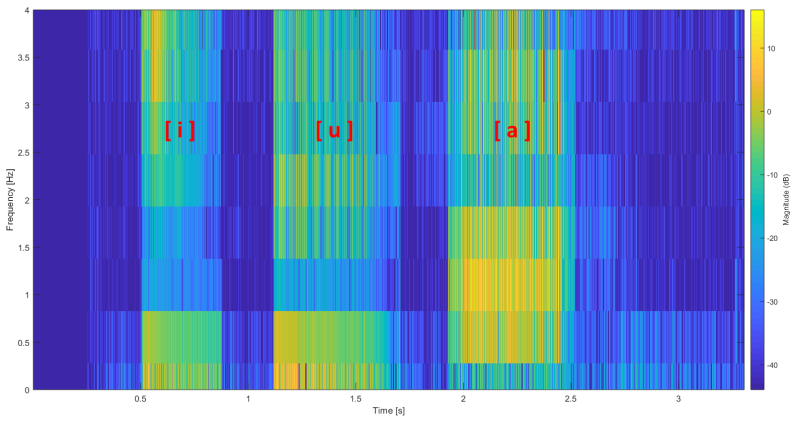


Figure 4.1: Speech formants for three Dutch vowels.

From the figure, different regions containing high spectral energy can be identified. It can be seen that for the vowel [i], most of the spectral energy is contained in frequency bands from 0-800 Hz and from 3000-4000 Hz. The vowel [u] has most of its spectral energy contained in frequency bands from 0-800 Hz and from 2000-2500 Hz, while the vowel [a] has most of its spectral energy contained in frequency bands from 250-1800 Hz and from 2500-3500 Hz. It is important to see that these formant bands are at different locations for different letters and thus change in time when considering a normal speech signal.

Speech formants are known to generally occur every 1000 Hz. This might be difficult to see in figure 4.1 however, since some of these speech formants can be relatively low in power or partially overlap with another speech formant.

When considering the spectrum of the complete speech signal instead of the spectrum of individual time-frames as in figure 4.1, these spectral energy bands can be more easily observed. This is shown in figure 4.2.

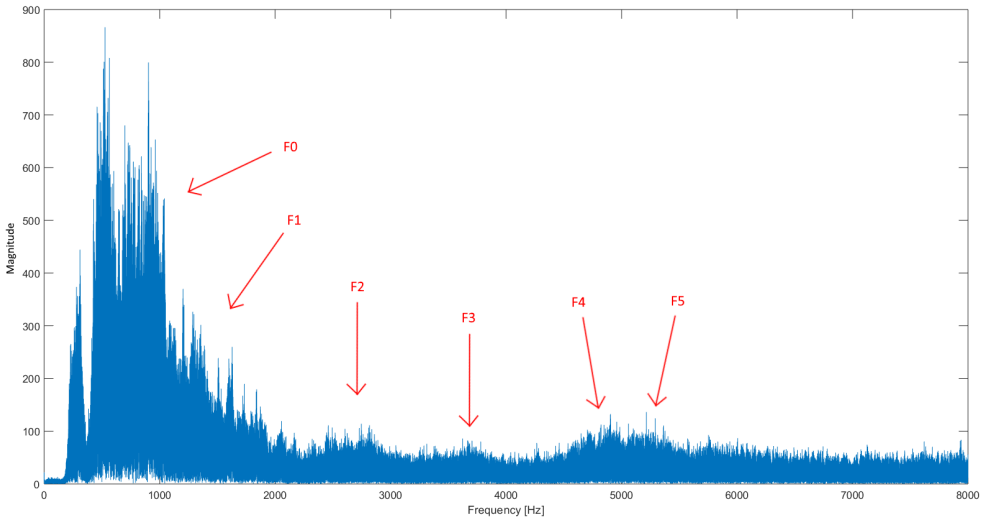


Figure 4.2: Spectrum with identified speech formants.

Here, the speech formants are numbered, starting from 0. The notion that speech formant locations change over time is lost in this figure, since we are looking at the whole speech signal.

5

Analysis of Human and Algorithmic Induced Modifications of Speech

In this chapter, different human and algorithmic induced speech modifications are considered and analyzed. The advantages and disadvantages in terms of the key performance indicators as described in chapter 3 are compared in order to select the relevant speech modifications for the *intelligibility enhancement subsystem.*

Speech modifications based on algebraic optimization for an objective intelligibility measure are considered in section 5.1. After that, human induced speech modifications are considered and discussed in section 5.2. Next, section 5.3 considers algorithmic induced speech modifications. Finally, the relevant speech modifications that will be implemented in chapter 6 are selected in section 5.4.

5.1. Measure Based Spectro-Temporal Energy Reallocation

Four different objective intelligibility measures have been briefly described in the state of the art analysis in section 1.2. With these objective measures, algebraic optimization of a speech signal becomes possible. When optimizing, the spectro-temporal energy of the speech signal is redistributed in such a way that it is more intelligible in the presence of near-end noise.

A requirement for this is that the near-end noise is known, this signal is provided by the *noise statistics estimation subsystem* as described in section 3.1. In an analysis of [5] and [6], it was found that spectro-temporal optimization comes down to defining a Lagrangian cost function, which can practically only be optimized using a bisection method. Therefore, there is no significant difference in time and space complexity and the overall system latency is the same for all mentioned measure-based optimizations (KPI 2, 3 and 4).

However, the correlation of these measures with word recognition rates in listening tests does differ. Therefore, it is important to select the proper objective intelligibility measure to optimize for, in order to maximize KPI 1.

Figure 5.1 shows part of the picture from [1], in which the relationship between word recognition in three different listening tests and objective intelligibility measures can be seen. No such data is available for SIIB^{Gauss}, but it is assumed to be similar to SIIB, since it is an approximation. The optimization methods as described below will be analyzed based on this figure.

The relevant information for these listening tests is shown in table 5.1.

Table 5.1: Listening test data set statistics as presented in [1].

Name	Degradation	Enhancement strategy	Test subjects	Test conditions
JensenMOD [20]	Modulated noise	None	12	60
KjemsITFS [21]	Noise	None	15	40
CookePRE [22]	Noise and competing talker	Enhancement algorithms	175	60

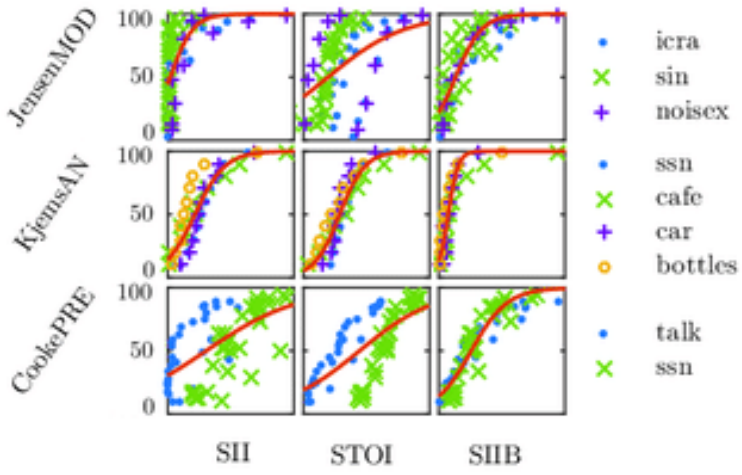


Figure 5.1: Scatter plots for three objective intelligibility measures as presented in [1]. The x-axis has been normalized to obtain a score between 0 and 1.

Spectro-Temporal Optimization for SII

The speech intelligibility index (SII), as proposed by the American National Standards Institute (ANSI) [2], is based on the idea that intelligibility is related to audibility [1]. A band-pass filterbank is applied to the clean speech signal and the near-end noise, the SNR per frequency band is then multiplied by a band importance scalar, clipped to ±15 SNR, normalized and summed to obtain the SII. This is shown in equation (5.1).

$$SII = \sum_i \gamma_i d(\xi_i). \tag{5.1}$$

In which i is the index of the frequency band, γ_i is the band importance scalar which is defined per frequency band, and $d(\xi_i)$ is the clipped and normalized SNR given by equation (5.2).

$$d(\xi_i) = \frac{max\Big(min\big(10log_{10}(\xi_i), 15\big), -15\Big)}{30} + \frac{1}{2}. \tag{5.2}$$

From equation (5.2), it can be seen that the SNR per frequency band is clipped to ±15 and normalized.

Taal et al. [5] has algebraically optimized for SII and concluded that optimizing for SII highly correlates with the word recognition rate in listening tests.

The relationship between this word recognition rate for three different listening tests and the SII can be seen in the first column of figure 5.1. It can be seen that the SII highly correlates with the word recognition for the JensenMOD and KjemsAN listening tests, while this is not so clear for the CookePRE listening test.

It is noted in Rhebergen et al. [23] that the SII performs well under non-fluctuating noise conditions, but not under fluctuating noise conditions.

Spectro-Temporal Optimization for STOI

Short time objective intelligibility (STOI) is an objective intelligibility measure proposed by Taal et al. [3]. Speech and speech degraded by near-end noise is time-frequency decomposed using 1/3 octave band decomposition and analyzed on short time-frames. The STOI is then calculated by means of the Pearson's correlation coefficient of the clean speech and degraded speech time-frames. This is shown in equation (5.3).

$$d = \frac{1}{JM} \sum_{j,m} d_{j,m}. \tag{5.3}$$

In which M represents the total number of frames, J the number of 1/3 octave bands, and $d_{j,m}$ the Pearson's correlation coefficient between the clean speech and degraded speech per frequency band and per time-frame.

The relation between word recognition and STOI is shown in figure 5.1. It can be seen that STOI highly correlates with the word recognition in the KjemsAN listening test, but not for the JensenMOD and CookePRE listening tests.

Spectro-Temporal Optimization for SIIB

The speech intelligibility in bits (SIIB) is proposed by Van Kuyk et al. [4] and is based on information theory. It is an objective measure for the mutual information between the clean speech signal and the speech signal degraded by near-end noise and is expressed in bits per second and is given in equation (5.4).

$$SIIB = \frac{F}{K} \sum_{j=1}^{KJ} min\Big(-\frac{1}{2} log_2(1 - r_j^2), I(\tilde{X}_j^k; \tilde{Y}_j^k) \Big). \tag{5.4}$$

In which F is the frame rate, K the number of stacked log spectra, r the production noise coefficient, j the eigenchannel index and $I(\tilde{X}_j^k; \tilde{Y}_j^k)$ is the mutual information between $\tilde{X}_j^k$ and $\tilde{Y}_j^k$. It is assumed that the production noise coefficient r is constant over all eigenchannels.

$\tilde{X}_j^k$ and $\tilde{Y}_j^k$ are vectors derived from the clean speech and degraded speech respectively, using a gammatone filterbank and the Karhunen–Loève transform (KLT).

From figure 5.1, it can be seen that SIIB highly correlates with the word recognition in all of the three considered listening tests.

A note is that the SIIB algorithm requires an input speech signal with a minimum voice active region of 20 seconds to function correctly.

Spectro-Temporal Optimization for SIIB$^{\text{Gauss}}$

SIIB can be simplified by assuming a Gaussian channel. This renders a KNN mutual information estimator [4], as used to calculate the SIIB, unnecessarily, thereby simplifying the computation.

This simplified version of SIIB is known as SIIB^{Gauss} and is proposed in [1]. Here, the mutual information is a function of the correlation coefficient only [24]. SIIB^{Gauss} is computed as in equation (5.5).

$$SIIB^{Gauss} = -\frac{F}{2K} \sum_j \log_2(1 - r^2 \rho_j^2). \tag{5.5}$$

In which F is the frame rate, K the number of stacked log spectra, r the production noise coefficient, j the eigenchannel index and ρ_j the correlation coefficient between the j^{th} clean eigenchannel an the j^{th} distorted eigenchannel.

Again, it is assumed that the production noise coefficient r is constant over all eigenchannels.

In figure 5.1, the correlation between the word recognition for different listening tests and the SIIB is shown. No such data is available for this approximation, but it is assumed that SIIB^{Gauss} has approximately the same correlation with the word recognition rates as SIIB.

The advantage of SIIB^{Gauss} is that it can be computed faster than SIIB, because there is no need for a KNN mutual information estimator as mentioned above. This computation time is two orders of magnitude less than the computation time of SIIB [1].

5.2. Human Induced Speech Modifications

When talking in situations in which it is difficult for the listener to understand the message conveyed in speech, humans automatically and involuntarily modify their speech in order to increase intelligibility for the listener. Two different effects can be distinguished based on two different situations in which intelligibility is degraded. First, there is the situation in which speech in subject to near-end noise. Human modified speech to overcome this problem is known as Lombard speech[8][9][25]. Second, there is the situation in which the listener is unable to fully understand the message conveyed in speech due to a hearing impairment. In this situation, human modified speech to overcome this problem is known as clear speech. Clear speech has also shown an increase in intelligibility for normal-hearing listeners in near-end noise [10].

Increased Sound Intensity (Lombard speech)

Summers et al. [9] has shown that when talking in noisy environments, humans exhibit a significant increase in speech intensity. This is the most profound effect of Lombard speech. An increase in sound intensity increases the intelligibility of a speech signal in a noisy environment by energetic masking of the noise because of an increased SNR, which logically leads to an increased intelligibility. This modification has as side-effects an increase in fundamental frequency F_0, an increase in vowel duration and a flattening of the spectral tilt [26].

Increase in Phonetic Fundamental Frequency F_0 (Lombard speech)

Increasing the fundamental frequency F_0, related to the first speech formant, the pitch of the speech signal will sound higher. However, an increase in F_0 has not been found to contribute to an increase of intelligibility in noisy environments according to [27], [28], [29], and [30]. The increase in range of F_0 also shows no increase in intelligibility [30].

Flattening of the Spectral Tilt (Lombard speech and clear speech)

The spectral density of speech signals will typically decrease for increasing frequency, this is known as spectral tilt and can be observed in figure 4.2. Flattening of the spectral tilt as observed in Lombard Speech, will increase energy in higher frequency bands and decrease energy in lower frequencies. It is effective in enhancing intelligibility in noise, as found in [31], [32] and [29]. The time complexity of this can be very low, since the flattening of the spectral tilt can be achieved with a Butterworth high-pass filter with a cut-off frequency f_c = 1.5 kHz [31]. The time complexity of such an algorithm is $O(nlogn)$. Davis et al. [33] implemented a 6 dB/octave high-pass filter that is used often for this application, for example in [11].

Increased Duration (Lombard speech and clear speech)

An intelligibility improvement is observed from a naturally decreased speech rate [34]. For Lombard speech, the duration of vowels is increased, while the overall signal duration is increased for clear speech. However, in [35] it was found that the artificial slowing of speech did not increase intelligibility in stationary noise, only in a fluctuating noise mask. The time complexity of such an algorithm can be as low as $O(n^2)$ when using a simple OLA (overlap-and-add) algorithm combined with vowel detection, while more extensive algorithms such as WSOLA (waveform similarity overlap-and-add) have a time complexity of $O(n^2 logn)$.

Shift in Formant Center Frequencies (Lombard speech)

A shift in formant center frequencies is a feature of Lombard speech. It modifies the location of the formant center frequencies. Nathwani et al. [36] synthesized this effect in order to improve the intelligibility of vocal sounds in a noisy car environment. An increase in intelligibility was found. This algorithm has a time-complexity of $O(n^5 log^3 n)$.

Increased consonant-vowel ratio (clear speech)

A redistribution of the energy such that the consonants are boosted with respect to vowels is called an increased consonant-vowel ratio. Gordonsalant et al. [37] found that increasing the consonant-vowel ratio by 10 dB showed an improvement in speech recognition. Skowronski et al. [31] implemented an algorithm that takes away energy from voiced regions, which are associated with vowels, and boosts unvoiced regions, which are associated with consonants, all while conserving global energy. This has shown to improve intelligibility in 9 out of 16 speakers without degrading the intelligibility of the other speakers. The time-complexity of this algorithm is $O(n)$.

5.3. Algorithmic Induced Speech Modifications

Enhancement of speech intelligibility can also be viewed from another perspective. Instead of mimicking the speech effects that humans exhibit, it is possible to investigate speech enhancements that are purely algorithmic induced and are shown to improve intelligibility, and not necessarily linked to human induced speech modifications.

Two of these algorithmic induced speech modifications are considered, namely dynamic range compression and transient amplification. These algorithms will be discussed below.

Dynamic Range Compression

Dynamic range compression is an alternate technique used to increase the consonant power compared to the power of vowels. Dynamic range compression in a situation where the speech signal is not noise-free, can amplify noise, since noise has a lower amplitude compared to speech. In this application, this is not a problem since the input speech signal is considered to be noise free (assumption 5). In [17] it was found in tests with speech shaped noise and competing speaker noise that the combination of spectral tilting and dynamic range compression leads to a great increase in intelligibility. The time-complexity of a feed-forward compressor is linear and therefore $O(n)$ [38].

Transient Amplification

Yoo et al. [12] has shown an increase in intelligibility for amplification of the transient component in a speech signal. Transient amplification algorithms all work in the same way in the sense that they extract the transient component of speech, amplify it and recombine the amplified transient component with the original speech signal. This is shown in equation (5.6).

$$\hat{s}(t) = \alpha\big(s(t) + \beta \cdot s_{trans}(t)\big). \tag{5.6}$$

In which β is the transient amplification factor and $0 < \alpha \leq 1$ is a coefficient to satisfy the power constraint as per mandatory requirement 3. A beneficial property of this algorithm is that the extraction of $s_{trans}(t)$ can be done independent of the noise statistics, facilitating pre-processing.

There are however different algorithms to extract this transient component, each with its own advantages and disadvantages regarding latency, time complexity, and word recognition rate improvement after recombination.

Time-Varying Band-pass Filters

In [13], Yoo et al. proposed an algorithm that extracts the transient component of speech using time-varying band-pass filters. This algorithm is based on the principle that the transient component exclusively exists of low spectral energy components. By removing the high spectral energy components from the input speech signal $s(t)$, only the transient component $s_{trans}(t)$ should remain.

According to Rasetshwane et al. [15], the duration of this algorithm is approximately 50 times the duration of the input speech signal $s(t)$. Therefore, this algorithm has a time complexity of $O(n)$, albeit with a high latency.

Static Filter

A static filter, as proposed by Rasetshwane et al. [15], is a static approximation of the time-varying band-pass filters by Yoo. A requirement is that the time-varying filter algorithm is executed at least once to have the enhanced speech signal $\hat{s}(t)$ available. After that, the static filter is calculated as described in equation (5.7).

$$H(e^{j\omega}) = \frac{|\hat{S}(e^{j\omega})|}{|S(e^{j\omega})|}. \tag{5.7}$$

After this is done, computing time is reduced since the input signal $s(t)$ does not have to be processed per time-frame, but can instead be calculated by convolution with the impulse response $h(t)$ as seen in equation (5.8).

$$\hat{s}(t) = s(t) * h(t). \tag{5.8}$$

Using fast convolution algorithms, the time complexity is $O(nlogn)$. However, the latency is significantly lower than the time-varying band-pass filter method.

Wavelet-Packet Transform
Tantibundhit et al. proposed a tonal and transient extraction algorithm based on the modified discrete cosine transform (MDCT) and the wavelet transform respectively [14]. The algorithm is able to determine coefficients for the wavelet transform in order to identify and extract the transient component. This algorithm will not be chosen for implementation, since the MDCT and the wavelet transform are not treated in the Bachelor program.

5.4. Selection of Feasible Methods

In order to make a selection of speech modifications that would suit the needs of the project as stated in section 1.1, a summary of the above mentioned methods in terms of key performance indicators is given in table 5.2.

Table 5.2: Summary of algorithmic and human induced speech modifications.

Method	Latency	TC (Big-O)	Intelligibility
Optimize SII	-	n^6log^2n	Not for fluctuating noise
Optimize STOI	-	n^5log^2n	Not for CookePRE test
Optimize SIIB	-	unknown	Great correlation
Optimize SIIB^{Gauss}	-	unknown	Assumed to have great correlation
Increase F_0	*not considered*		No increase
Reduced spectral tilt	+	nlogn	Best combined with CV-ratio
Increased sound intensity	++	1	In all conditions
Increased vowel duration	-	n^2	Only in non-stationary noise
Shift formant center frequencies	–	n^5log^3n	
Increased consonant-vowel ratio	+	n	Best combined with spectral tilt
Dynamic range compression	-	n	Good when combined with tilt
Time-varying band-pass filters	-	n	Shown to have good results
Static filter	+	$nlogn$	Approximation of time-varying
Wavelet-packet transform	*not considered*		

Based on this analysis, one algorithm will be chosen based on measure based spectro-temporal energy reallocation. Several features based on human induced speech modifications and one algorithm to extract the transient component of speech, including a static approximation, will be chosen for implementation.

When considering measure based spectro-temporal energy reallocation, it is important to optimize for a measure that highly correlates with word intelligibility in a listening test in which conditions are similar to the problem in section 1.1.

When comparing the listening test conditions in table 5.1, it can be seen that the CookePRE test has 175 test subjects, whereas the other two test only have 12 and 15. Apart from that, speech in the CookePRE test is degraded with speech-shaped noise and competing talker noise, which corresponds to the problem stated in section 1.1. Therefore it is important to optimize for a measure that highly correlates with data from the CookePRE test.

As can be seen from the analysis, this is only the case for SIIB and SIIB^{Gauss}. Out of these two, SIIB^{Gauss} is selected for implementation because of the decreased computation time compared to SIIB.

Considering human induced speech modifications, only those modifications that are shown to have a positive effect on the intelligibility of speech are chosen for implementation. The choice was made to not include

an implementation of the shifting of formant center frequencies, since the time-complexity is large compared to the other Lombard and clear speech modifications. The features that are chosen to be implemented are an increased vowel duration and a reduced spectral tilt, which will be combined with dynamic range compression because it is shown to produce great results. The increased sound intensity will be implemented in the form of determining the necessary amplification factor that the *amplifier and noise cancellation subgroup* will use to amplify the signal. From here on, the human induced speech modification algorithms will be referred to as Lombard algorithm.

Finally, considering algorithmic induced speech modifications, dynamic range compression is combined with the implementation of Lombard speech, as an alternative for increase consonant-vowel ratio. This leaves transient amplification based on two different transient extraction algorithms for consideration, since the wavelet-packet transform based algorithm is not considered for implementation.

The time-varying band-pass filters and the static filter implementation are both selected for implementation. While the time complexity of the static filter method is higher than that of the time-varying filter method, the latency is significantly lower. Both are shown to have a positive effect on the intelligibility of speech.

Because optimisation for SIIB^{Gauss} is selected for the implementation of the measure based spectro-temporal energy reallocation method, it would only seem logical to asses the working of the other algorithms to be implemented based on SIIB^{Gauss} as well. Therefore, all modifications implemented in chapter 6 are assessed based on SIIB^{Gauss}.

6

Implementation of Speech Modification Algorithms

In chapter 5, the different speech modification methods, be it human- or algorithmic induced, were discussed and analyzed. Based on this analysis and the key performance indicators and requirements as stated in chapter 3, three different implementations were selected.

First, the algebraic optimization for SIIB^{Gauss} is implemented in section 6.1. This will result in an enhanced speech signal $\hat{s}(t)$ that has optimal intelligibility in the presence of near-end noise according to SIIB^{Gauss}.

Next, an algorithm based on the effects observed in Lombard speech is implemented in section 6.2. From chapter 5 it became clear that increasing the vowel duration and decreasing the spectral tilt combined with compression of the dynamic range are effects of Lombard speech that satisfy the proposed project in terms of key performance indicators and project requirements.

Finally, an algorithm based on the amplification of transient component of speech is implemented in section 6.3. This algorithm is based on time-varying band-pass filters as proposed by Yoo et al. [13]. A static approximation of this amplification method based on time-varying filters is also implemented, since it is shown in chapter 5 to greatly reduce computation time.

All of the Matlab code for these implementations can be found in appendix B.

6.1. Optimize for SIIB^{Gauss}

SIIB^{Gauss} is an objective intelligibility measure that highly correlates with the word recognition rate in listening tests and is suitable for algebraic optimization. The equation for SIIB^{Gauss} is given in equation (5.5) and is for convenience repeated in equation (6.1).

$$SIIB^{Gauss} = -\frac{F}{2K}\sum_{j}\log_2(1-r^2\rho_j^2). \tag{6.1}$$

In which F is the frame rate, K the number of stacked log spectra, r the production noise coefficient, j the eigenchannel index and ρ_j the correlation coefficient between the j^{th} clean eigenchannel an the j^{th} distorted eigenchannel. It is assumed that the production noise coefficient r is constant over all eigenchannels.

Writing the $log_2(x)$ as a function of $ln(x)$, as in equation (6.2), equation (6.1) can be rewritten to equation (6.3).

$$SIIB^{Gauss} = -\frac{F}{2K}\sum_{j}\frac{ln(1-r^2\rho_j^2)}{ln(2)}, \tag{6.2}$$

$$SIIB^{Gauss} = -\frac{F}{2Kln(2)}\sum_{j}ln(1-r^2\rho_j^2). \tag{6.3}$$

Using a first order Taylor series expansion, the $ln(x)$ in the sum of equation (6.3) can be approximated by equation (6.4) and equation (6.6) is obtained by taking the production noise coefficient r out of the sum in equation (6.5) and grouping it as a constant.

$$ln(1-r^2\rho^2) \approx -r^2\rho^2, \tag{6.4}$$

$$SIIB^{Gauss} \approx -\frac{F}{2Kln(2)}\sum_j -r^2\rho_j^2, \tag{6.5}$$

$$SIIB^{Gauss} \approx C\sum_j \rho_j^2. \tag{6.6}$$

From equation (6.6) it can be seen that optimizing for SIIB^{Gauss} comes down to optimizing for the squared correlation coefficient ρ_j^2. This makes sense, since SIIB^{Gauss} is a measure using the mutual information between the clean speech X and the by near-end noise distorted speech $Y = X+N$. The mutual information is at a maximum when Y contains all the information contained in X, meaning these signals are highly correlated.

The correlation coefficient ρ_j^2 between signals x_j, the j^{th} eigenchannel of X, and y_j, the j^{th} eigenchannel of Y, is given in equation (6.7), where σ_{x_j} σ_{y_j} are the standard deviation of x_j and y_j respectively and $Cov(x_j, y_j)$ is the covariance between the two signals. Since x_j and y_j are zero-mean signals, the covariance is equal to the expectation of $x_j y_j$, namely $E[x_j y_j]$.

$$\rho_j^2 = \frac{Cov(x_j,y_j)^2}{\sigma_{x_j}^2\sigma_{y_j}^2} = \frac{E[x_j y_j]^2}{\sigma_{x_j}^2\sigma_{y_j}^2}. \tag{6.7}$$

For simplicity, it is assumed that these eigenchannels can be considered equal to the frequency bins as used by gammatone filterbanks.

The distorted signal $Y = X + N$ is the sum of the clean signal and the noise, which are assumed to be uncorrelated (assumption 1). Therefore, equation (6.7) can be rewritten to equation (6.8).

$$\rho_j^2 = \frac{E[x_j(x_j+n_j)]^2}{\sigma_{x_j}^2(\sigma_{x_j}^2+\sigma_{n_j}^2)} = \frac{(E[x_j^2]+E[x_j n_j])^2}{\sigma_{x_j}^2(\sigma_{x_j}^2+\sigma_{n_j}^2)}. \tag{6.8}$$

In which $E[x_j n_j] = 0$ because x_j and n_j are independent and zero-mean signals. This is used to rewrite equation (6.8) into equation (6.9).

$$\rho_j^2 = \frac{E[x_j]^2}{(E[x_j^2]-E[x_j]^2)(E[x_j^2]-E[x_j]^2+\sigma_{n_j}^2)} = \frac{E[x_j]^2}{(E[x_j^2]^2-2E[x_j]^2E[x_j^2]+E[x_j^2]\sigma_{n_j}^2+E[x_j]^4-E[x_j]^2\sigma_{n_j}^2)}. \tag{6.9}$$

Since the expectation of x_j is zero, this simplifies into equation (6.10).

$$\rho_j^2 = \frac{E[x_j^2]^2}{E[x_j^2]^2+E[x_j^2]\sigma_{n_j}^2} = \frac{1}{1+\frac{\sigma_{n_j}^2}{E[x_j^2]}}. \tag{6.10}$$

Using $E[x_j^2] = \sigma_{x_j}^2$ and $\frac{\sigma_{n_j}^2}{\sigma_{x_j}^2} = \xi^{-1}$, equation (6.11) is obtained.

$$\rho_j^2 = \frac{1}{1+\xi^{-1}}, \tag{6.11}$$

$$\rho_j^2 = \frac{\xi}{1+\xi}. \tag{6.12}$$

From equation (6.6) it can be seen that optimizing for $SIIB^{Gauss}$ is equivalent to optimizing equation (6.12). The correlation coefficient squared can be optimized by applying the Karush-Kuhn-Tucker conditions to the Lagrangian cost-function, equation (6.13), where $r = \sum_i \sigma_{X_i}^2$, as done by Taal et al. [5]. Any point that satisfies this conditions is guaranteed to be optimal [39].

$$J = -\sum_j \frac{\alpha_j^2 \xi_j}{\alpha_j^2 \xi_j + 1} + v(\sum_j \alpha_j^2 \xi_j - r) + \sum_j \lambda_j(-\alpha_j^2 \sigma_{X_j}^2). \tag{6.13}$$

Two constraints are that the energy of the enhanced signal is equal to the energy of the original signal (mandatory requirement 3) and that all coefficients are greater than or equal to zero. This is shown in equation (6.14) and equation (6.15) respectively.

$$\sum_j \alpha_j^2 \sigma_{X_j}^2 = \sum_j \sigma_{X_j}^2, \tag{6.14}$$

$$\alpha_i^2 \sigma_{X_j}^2 \geq 0, \forall j. \tag{6.15}$$

Solving the Lagrangian cost-function in equation (6.13) gives equation (6.16).

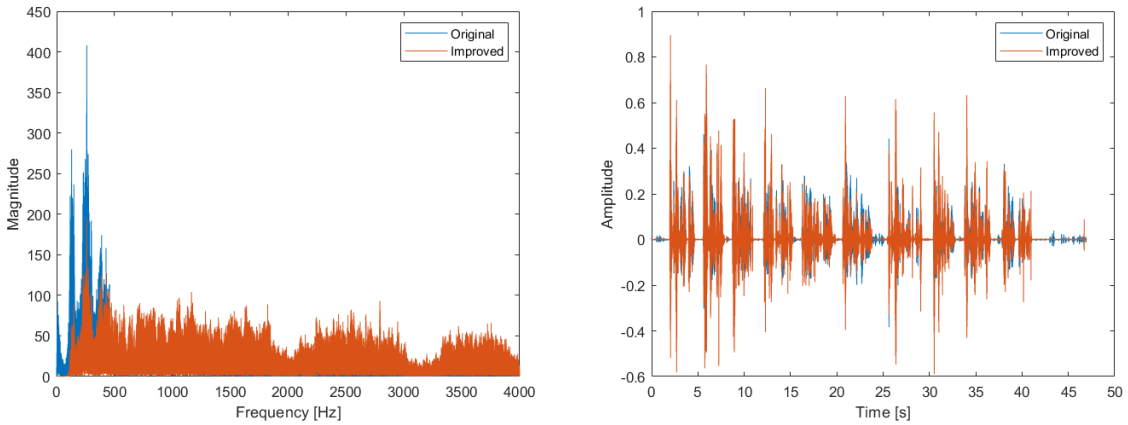
$$\alpha_j^2 \sigma_{X_j}^2 = max(\frac{\sigma_{n_j}}{\sqrt{v}} - \sigma_{n_j}^2, 0), \forall j. \tag{6.16}$$

In equation (6.17), v is chosen such that the energy constraint in equation (6.14) is satisfied.

$$\frac{1}{\sqrt{v}} = \frac{r + \sum\limits_{i \in M} \sigma_{n_j}^2}{\sum\limits_{i \in M} \sigma_{n_j}^2}. \tag{6.17}$$

In which M denotes the set of critical bands in which the optimal α_j^2 is positive. This means that the Lagrange multiplier v is also dependent on α_j^2, which gives rise to a recursive dependency in equation (6.16). Therefore a bisection method will have to be used in order to determine the optimal value of v [5]. The values α for each frequency bin are then calculated using equation (6.16). Finally, α is multiplied with the frequency decomposed signal and transformed back to the time domain.

A one-sided frequency spectrum of a signal for which the energy is redistributed using the described SIIB^{Gauss}-optimization is shown in figure 6.1a and the time-domain signal is given in figure 6.1b.



(a) Frequency spectrum of original and enhanced signal.

(b) Original and enhanced signal in time domain.

Figure 6.1: Example of enhanced signal using the SIIB^{Gauss}-optimization.

The improvement in intelligibility in terms of SIIB^{Gauss} using this optimization algorithm is given in table 6.1 for an SNR of 3 dB and in table 6.2 for an SNR of -4.7 dB.

Table 6.1: Intermediate results in terms of SIIB^{Gauss} of the SIIB^{Gauss}-optimization for SNR = 3 dB.

	Non-fluctuating noise	Fluctuating noise
Original	76.5 [bits/s]	90.1 [bits/s]
Improved	136 [bits/s]	159.9 [bits/s]

Table 6.2: Intermediate results in terms of SIIB^{Gauss} of the SIIB^{Gauss}-optimization for SNR = -4.7 dB.

	Non-fluctuating noise	Fluctuating noise
Original	12.8 [bits/s]	12.5 [bits/s]
Improved	16.6 [bits/s]	12.3 [bits/s]

In an SNR of 3 db, the increase in intelligibility is substantial in both non-fluctuating and fluctuating noise, while in a much lower SNR of -4.7 dB, only in non-fluctuating noise an increase is observed.

6.2. Lombard Effect

Three speech modifications that will be implemented are increased vowel duration, flattening of the spectral tilt and dynamic range compression. In chapter 5.2, these modifications were selected based on their performance with regard to the KPI's of the system.

6.2.1. Vowel Duration

An increase in vowel duration will increase the duration of the overall signal, therefore a power constraint in which the power of the elongated signal is equal to the power of the input speech signal, is not appropriate here. Rather, a normalization method relative to the duration of the input and output signals is considered, as stated in chapter 2. This is described in equation 6.18.

$$\sum_{j=1}^{m} |Y_j| = \frac{D_y}{D_x} \cdot \sum_{i=1}^{n} |X_i|. \tag{6.18}$$

Where Y is the Fourier transform of the Lombard processed signal y, m is the length of Y, X is the Fourier transform of the input speech signal x, n is the length of X and D_y and D_x are the duration of the Lombard and input speech signals respectively.

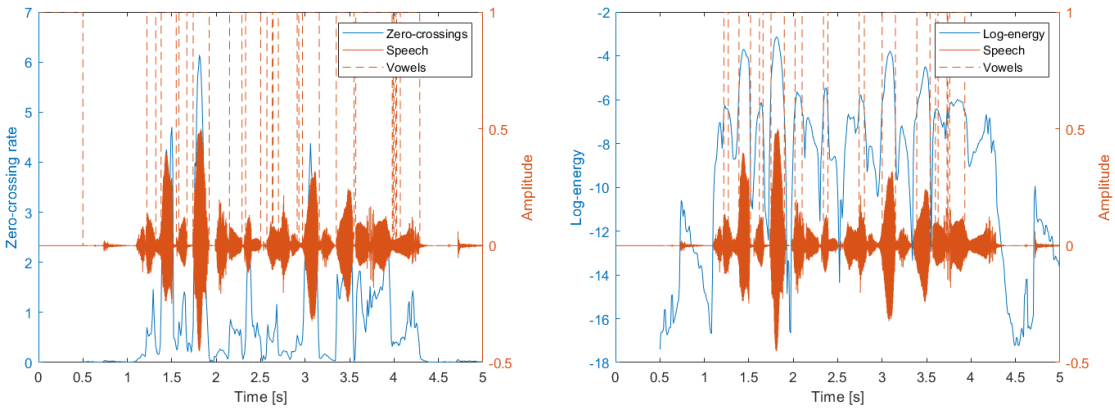
In order to increase the vowel duration, the vowels need to be located in the input speech signal. For this, the speech signal $s(n)$ is divided in short time-frames as shown in equation (6.19).

$$s_l(n) = s(n + lR)\, w(n), n = 0, ..., N-1. \tag{6.19}$$

Where R is the hop length of 10 ms, N the frame length of 20 ms, $w(n)$ the rectangular windowing function and l the frame index. There are several methods to locate vowels within a speech signal [40]:

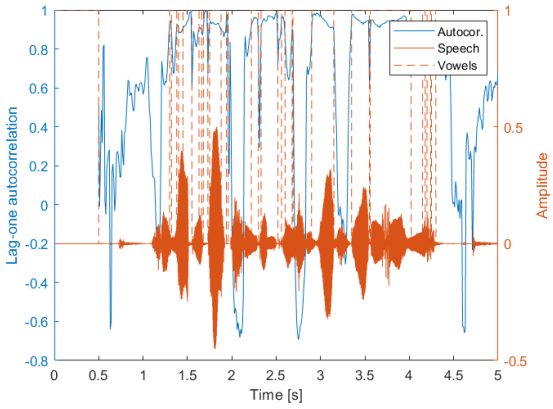
- Zero-crossing Rate
  - $ZCR_l = \sum_{n=1}^{N} I(s_l(n)s_l(n-1) < 0)$ where $I(a < 0) = 1$ and 0 otherwise. The zero-crossing rate is higher in consonants than in vowels.
- Log-energy:
  - $E_{s,l} = log(\frac{1}{N}\sum_{n=0}^{N} s_l^2(n))$. The log-energy of vowels is greater than consonants.
- Lag-one autocorrelation
  - $\gamma_l(1) = \frac{\sum_{n=1}^{N} s_l(n)s_l(n-1)}{\sqrt{\sum_{n=1}^{N} s_l^2(n)\sum_{n=0}^{N-1} s_l^2(n-1)}}$, where voiced sounds are highly correlated and unvoiced sound are not.

These methods were implemented and the results are shown in figure 6.2.



(a) Zero-crossing rate.

(b) Log-energy.



(c) Lag-one Autocorrelation.

Figure 6.2: Different vowel detection algorithms.

The zero-crossing rate technique should be high for consonants and low for vowels. In figure 6.2a, it can be seen that this is not consistently true. The lag-one autocorrelation should be high at vowels, and drop at consonants. In figure 6.2c, it can be seen that this is mostly true, but that the difference is small. A drop occurs at the end of the fragment, making it difficult to set a constant threshold for this implementation. It can be seen that the log-energy method in figure 6.2b gives the best results for detecting the vowels, so that implementation was chosen to use for vowel recognition.

In [41], a review of time-scale modification algorithms is given. The algorithms OLA (overlap and add), WSOLA (waveform similarity overlap and add) and a phase vocoder are compared. The OLA algorithm was found to be not suited to modify signals that contain harmonic components due to phase jump artifacts. The output signals of the phase vocoder often suffer from phasiness, which is described as a 'loss of presence'. This is not desirable for this application. WSOLA causes an artifact that is known as transient doubling or stuttering, but since in this application only vowels are stretched, it is well suited. However, WSOLA is a more complex algorithm, which will increase the latency. For this reason, OLA was chosen.

Applying the OLA algorithm combined with vowel detection on a clean speech signal, a signal with stretched vowels is obtained, as demonstrated in figure 6.3. Optimizing the vowel duration can not be done using SIIB^{Gauss}. No optimum can be found using this, as can be seen in figure 6.4a, so informal listening tests are used, as described in appendix A. In figure 6.4b the word recognition rate for different vowel durations can be seen in non-fluctuating and in fluctuating noise. From this, an optimal vowel stretch factor of 2.5 is found in non-fluctuating noise and a stretch factor of 1.9 in fluctuating noise.

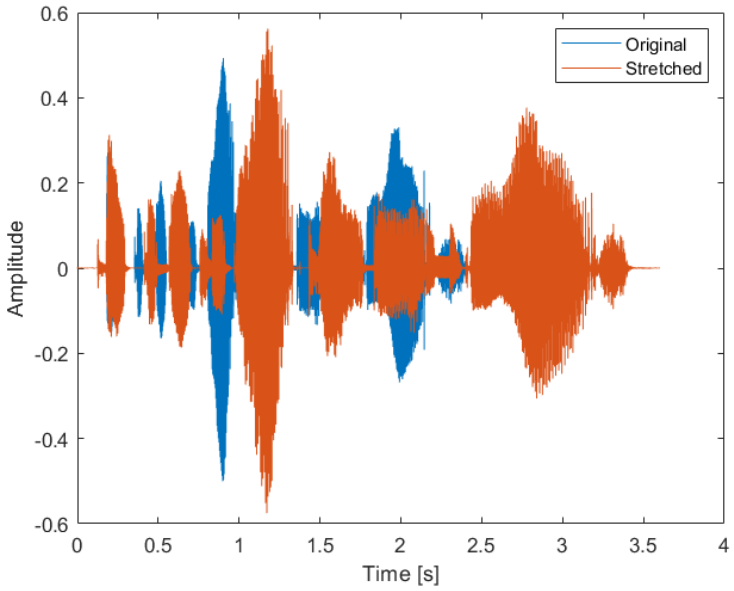
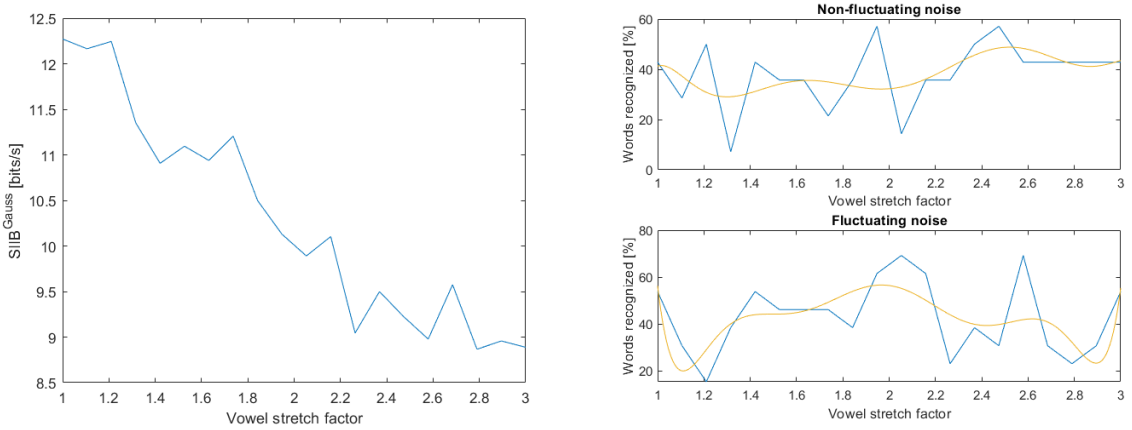


Figure 6.3: Clean speech signal of which the vowels are stretched with a factor of 2. The non-stretched signal is given in blue and the modified signal in red.



(a) SIIB value for different vowel stretch factors.

(b) Listening test for different vowel stretch factors.

Figure 6.4: Analysis of filter coefficients at an SNR of -4.7 dB.

Using these optimal values, another informal listening test was done to determine the increase in intelligibility. The results from this are seen in table 6.3. As can be seen, in both noise conditions the word recognition rate is reduced. It is believed that this is because of artifacts introduced by vowel stretching. Another possible cause is that the test that was used was not suitable for the algorithm, since the difference between the words in one word-set was one consonant, meaning vowelstretching would not improve the word recognition.

Table 6.3: Vowel stretching listening test; word recognition rate.

	Non-fluctuating noise	Fluctuating noise
Original	79.1 %	61.7 %
Improved	64.1 %	57.5 %

6.2.2. Spectral Tilting

In chapter 4, it is described that different formants can be observed in speech. From figure 6.5 it can be observed that most of the power is contained in the first formant F_0, while higher formants (F_1, F_2 and F_3) contain less power. This is also shown in figure figure 4.2. This gives rise to a steep spectral tilt in which lower frequencies contain more power than high frequencies.

By reducing the spectral tilt, higher frequencies will effectively receive more power while the power of the first formant F_0 is reduced. This is done by implementing a high-pass filter with transfer function equation (6.20) and then normalization of the energy. The spectrum of the decreased spectral tilt is shown in figure 6.5.

$$H(z) = \frac{1}{1 - az^{-1}}. \tag{6.20}$$

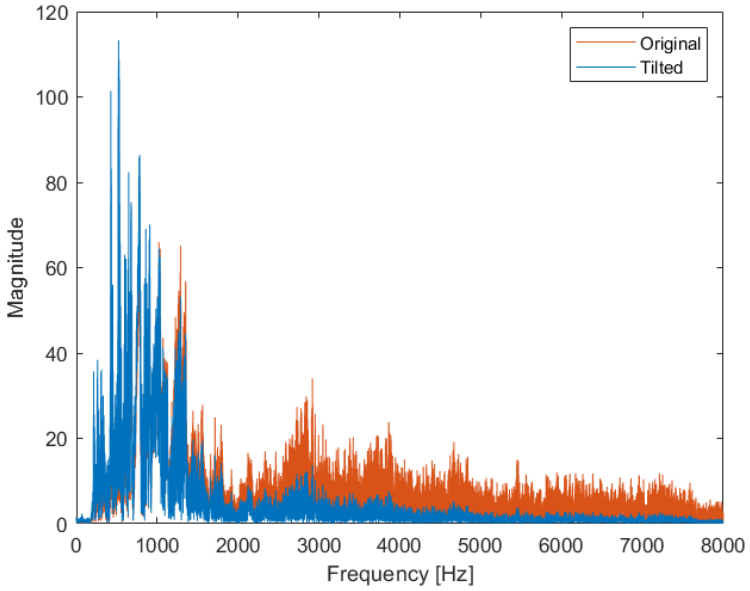
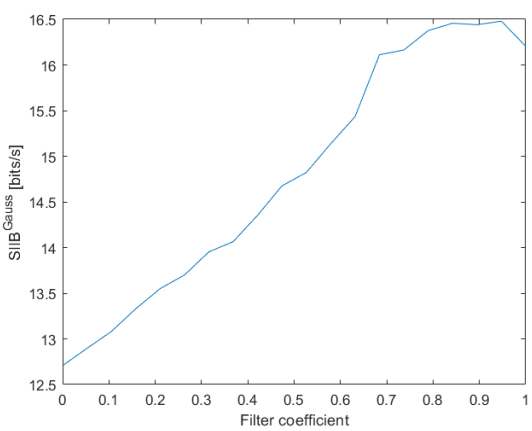


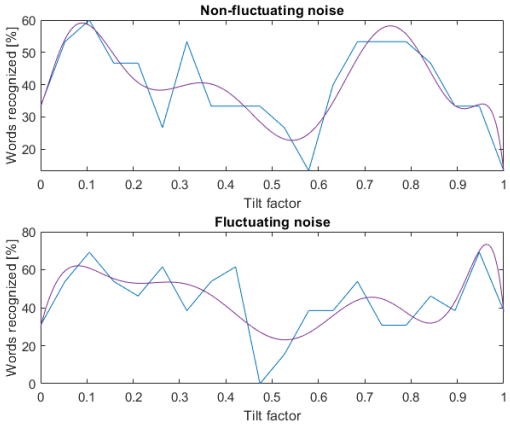
Figure 6.5: Frequency spectrum of spectral tilting with a filter coefficient of 0.8.

Since reducing the spectral tilt does not alter the duration of the input speech signal, D_y is equal to D_x and equation (6.18) reduces to equation (6.21).

$$\sum_{j=1}^{m} |Y_j| = \sum_{i=1}^{n} |X_i|. \tag{6.21}$$



(a) SIIB value for different filter coefficients.



(b) Listening test for different filter coefficients.

Figure 6.6: Analysis of filter coefficients at an SNR of -4.7 dB.

Using the SIIB^{Gauss} values from figure 6.6a and the informal listening test results in figure 6.6b, optimal values for the filter parameter a in equation (6.20) were found to be 0.77 in non-fluctuating noise and 0.95 in fluctuating noise. Using these values, another informal listening test was done to assess the intelligibility of the enhanced signal. The results of this test are given in table 6.4. In stationary noise, a decrease in intelligibility is found, however in fluctuating noise the word recognition is approximately equal.

Table 6.4: Spectral tilt listening test; word recognition rate.

	Non-fluctuating noise	Fluctuating noise
Original	62.5 %	73.3 %
Improved	45.8 %	74.1 %

6.2.3. Dynamic Range Compression

In [17] and [16], flattening of the spectral tilt is combined with dynamic range compression. Dynamic range compression moves the energy from high energetic vowels to less energetic consonants. An example of a compressed audio signal is shown in figure 6.7. The compressor from Matlab's audio toolbox is used for this implementation [42][38].

In figure 6.8a, it can be seen that the SIIB^{Gauss} value increases for an increasing negative threshold. When listening to such a signal, it an be observed that the sound quality degrades greatly with such a threshold. This is possibly due to the amplification of imperfections in the speech signal, which were too quiet to be heard at first. Therefore, an informal listening test was done, the results from these tests are given in figure 6.8b. The optimal values for the threshold were found to be -8.8 dB in stationary noise and -1.5 dB in fluctuating noise.

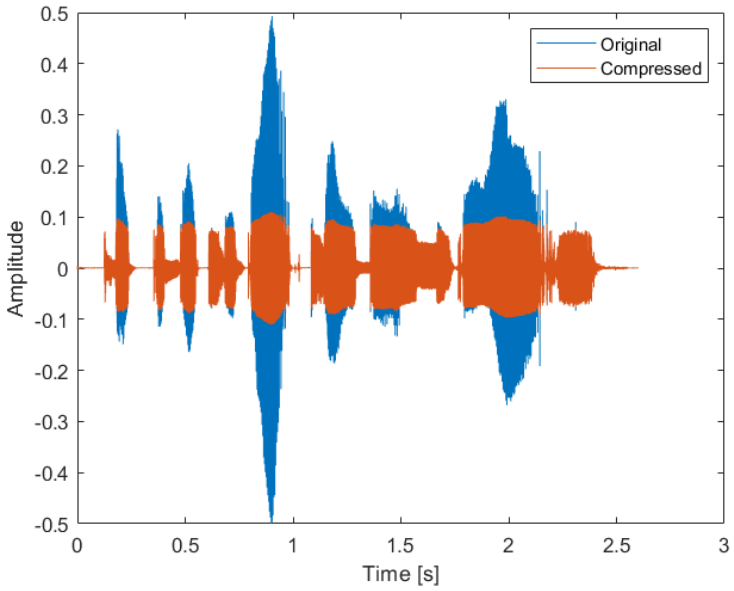
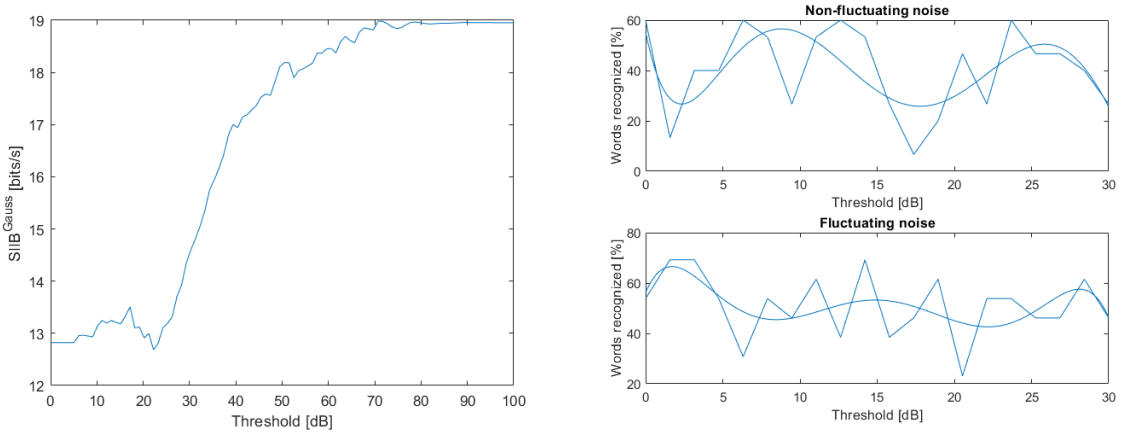


Figure 6.7: Dynamic range compression with a threshold of -30 dB.



(a) SIIB value for different thresholds.

(b) Listening test for different thresholds.

Figure 6.8: Analysis of threshold levels at an SNR of -4.7 dB. The absolute values of the threshold are plotted on the x-axis.

With these optimal values, another listening test was done to assess the improvement in intelligibility. The results are given in table 6.5. Both in stationary and in fluctuating noise an improvement in word recognition is observed.

Table 6.5: Dynamic Range Compression listening test; word recognition rate.

	Non-fluctuating noise	Fluctuating noise
Original	46.7 %	74.1 %
Improved	56.7 %	77.5 %

6.3. Transient Amplification

An intelligibility enhancement algorithm is implemented based on the transient amplification method using time-varying band-pass filters, as proposed by Yoo et al. [13].

It is shown by Daudet and Torrésani [43] that speech signals can be modeled by the superposition of a tonal, a transient, and a residual component. This model can be seen in equation (6.22).

$$s_{speech}(t) = s_{tonal}(t) + s_{transient}(t) + s_{residual}(t). \tag{6.22}$$

In which $s_{speech}(t)$ is the complete speech signal, $s_{tonal}(t)$ is the tonal component of speech, $s_{transient}(t)$ is the transient component of speech, and $s_{residual}(t)$ is the residual component.

It is shown by Yoo et al. [12] that this transient component of speech greatly contributes to the intelligibility of the speech signal, in contrast to the tonal component. However, this transient component only contains about 2% of the total energy of the speech signal [12]. If this transient component is amplified and recombined with the original speech signal, an increase in intelligibility can be observed [44] [13].

A constraint is that the power of the enhanced signal is equal to the power of the original speech signal as described in mandatory requirement 3. If not for this power constraint, the trivial case would be considered in which intelligibility is increased by improving the SNR.

This intelligibility enhancement algorithm based on transient amplification as described above is shown in equation (6.23).

$$\hat{s}(t) = \alpha\big(s(t) + \beta \cdot s_{transient}(t)\big). \tag{6.23}$$

In which $\hat{s}(t)$ is the enhanced speech signal, $0 < \alpha < 1$ is a scalar to satisfy the power constraint, β is the amplification factor of the transient component, and $s_{transient}(t)$ is the transient component of the input speech signal $s(t)$.

First, the transient part of speech needs to be estimated before it can be amplified. In chapter 5 different algorithms for the estimation of the transient component of speech are considered. Based on this analysis, the time-varying band-pass filter algorithm by Yoo et al. [13] and the static filter approximation of this method by Rasetshwane et al. [15] are selected for implementation and will be implemented in section 6.3.1 and section 6.3.2 respectively.

6.3.1. Time-Varying Band-pass Filter Implementation

Since the transient component of speech only contains a small amount of power [12], band-pass filters can be used to extract the high powered formant signals and subtract them from the speech signal. Because the positions of the speech formants change with time, as can be observed in figure 4.1, these band-pass filters should be time-varying. The residual of the above subtraction should then be equal to the transient component of speech. This is shown in equation (6.24).

$$s_{transient}(t) = s(t) - \sum_{i=0}^{F} f_i(t). \tag{6.24}$$

In which F is the total number of speech formants identified in one time-frame of the speech signal and $f_i(t)$ is the extracted formant signal of the i-th formant. $s(t)$ and $s_{transient}(t)$ denote the speech signals in a single time-frame.

The length of the time-frames is selected in such a way that the lowest frequency component of speech has at least one complete period in the time-frame, in order for it to be correctly recognized. Generally, 20 Hz is considered to be the lowest frequency to which the human ear is perceptive. The period of a 20 Hz signal is 50 ms. To be on the safe side and to reduce computation time, a length of 100 ms is selected for the time-frames.

The formant signal $f_i(t)$ is extracted by a band-pass filter that is placed around the center frequency of the formant band. This can be seen in figure 6.9.

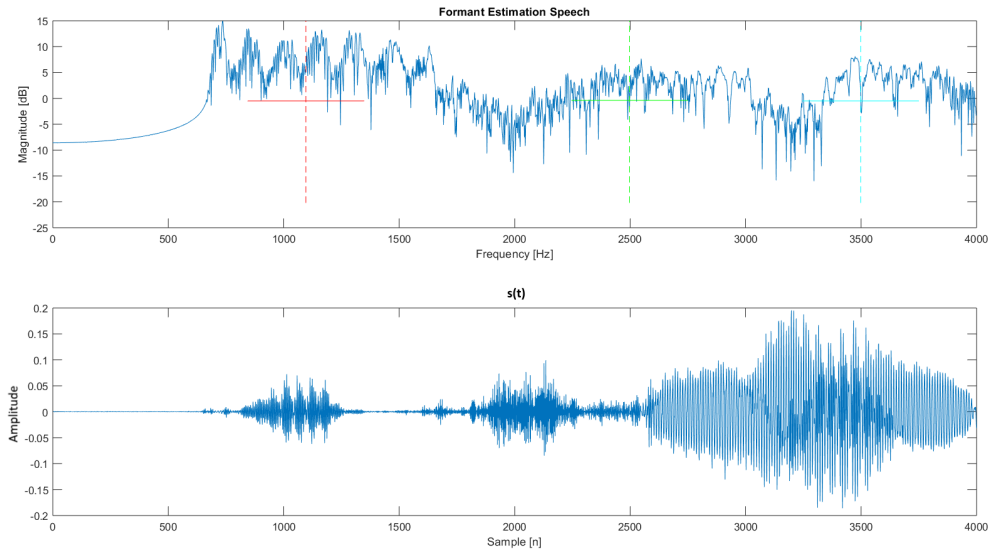


Figure 6.9: Filter design for formant removal in individual time-frames.

The dashed line in figure 6.9 represents the estimated formant center frequency. This center frequency is estimated by the frequency bin with the highest spectral power within the typical range for that formant. Since formants occur typically every 1000 Hz, the center frequency of $F1$ is assumed to be between 1000 Hz and 2000 Hz.

A band-pass filter with a predefined range is placed around these center frequencies. When these band-pass filters overlap, these bounds are adjusted to assure the same frequencies are not subtracted twice in equation (6.24). In the case where the upper bound of the third formant band-pass filter (the cyan line in figure 6.9) is close to halve the sampling frequency, a high-pass filter is used. This can be seen in figure 6.10a and figure 6.10b respectively.

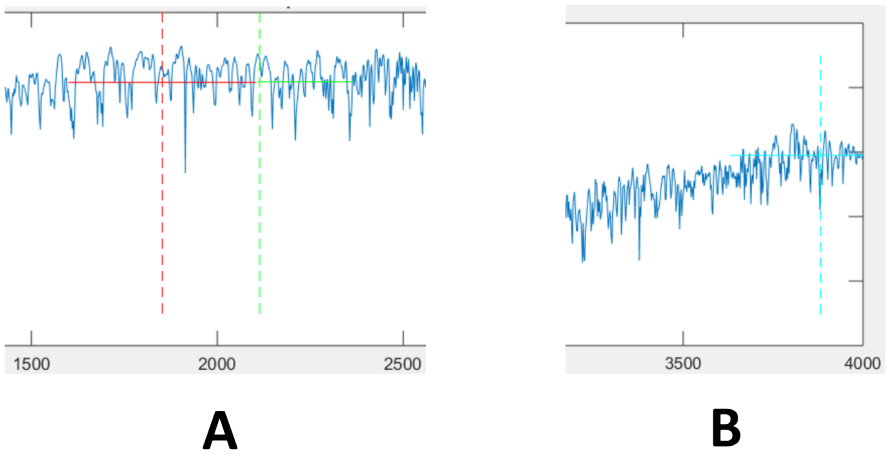
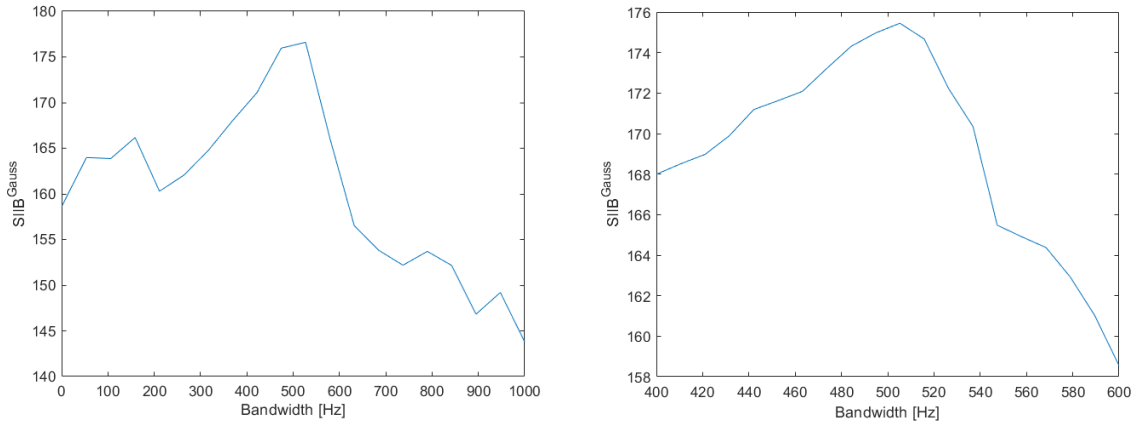


Figure 6.10: A) Band-pass filter overlap. B) High pass filter required because upper bound exceeds half the sampling frequency.

Once all the speech formants are estimated per time-frame, the transient component is calculated using equation (6.24) and the enhanced speech signal $\hat{s}(t)$ can be obtained using equation (6.23).

To determine the optimal band-pass filter width and amplification factor β in equation (6.23), this enhanced speech signal is analyzed in terms of the SIIB^{Gauss}. It is assumed that these two parameters can be optimized independently.

First, the band-pass filter width is optimized for SIIB^{Gauss}. This is done for a constant amplification factor $\beta = 12$ in equation (6.23). The resulting SIIB^{Gauss} for 20 different band-pass widths between 1 Hz and 1000 Hz are shown in figure 6.11a.



(a) Analysis of formant removal filter bandwidth.

(b) Analysis of formant removal filter bandwidth (zoom).

Figure 6.11: Analysis of formant removal filter bandwidth.

From this picture, it becomes clear that there is an optimal value for the width of the band-pass filters between 400 Hz and 600 Hz. A more detailed picture of 20 linearly spaced band-pass widths between 400 and 600 Hz can be found in figure 6.11b.

From figure 6.11b, a bandwidth of 505 Hz is selected as the optimum band-pass filter bandwidth for the transient extraction algorithm.

Next, the transient amplification algorithm is again analyzed for SIIB^{Gauss}, but this time an optimal value for the transient amplification factor β in equation (6.23) is estimated. It has to be taken into account that the signal to noise ratio can influence the value of the optimal amplification factor.

In figure 6.12, the analysis in terms of SIIB^{Gauss} of the transient amplification algorithm can be seen under different SNRs and 20 linearly spaced amplification factors between 0 and 20 are considered.

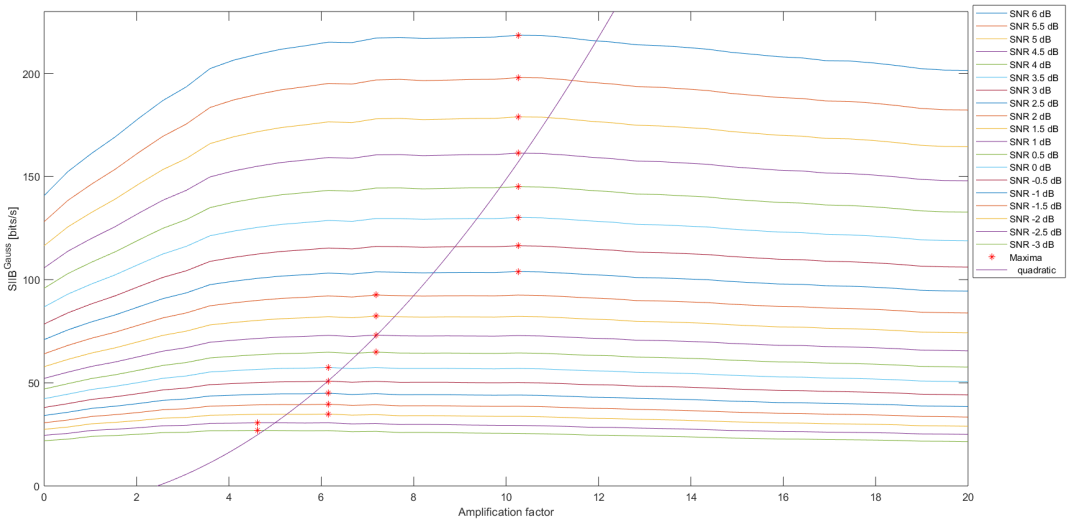


Figure 6.12: Analysis of transient amplification factor for different SNRs.

From the figure it can be seen that a greater increase in SIIB^{Gauss} can be obtained in higher SNRs. It can also be seen that the optimal amplification factor depends on the SNR, however, since the intelligibility in terms of SIIB^{Gauss} does not change significantly close to the optimal amplification factor, an estimate of $\beta = 10$ can be used as the optimal amplification factor without significantly degrading the intelligibility at lower SNRs.

The intermediate results obtained with the transient amplification algorithm from equation (6.23) based on the time-varying band-pass filter method can be found in table 6.6 and table 6.7, in which high SNR and low SNR conditions are considered respectively.

Table 6.6: Intermediate results in terms of SIIB^{Gauss} for transient amplification SNR = 3 dB.

	Non-fluctuating noise	Fluctuating noise
Original	76.6 [bits/s]	91.2 [bits/s]
Improved	114.6 [bits/s]	132.2 [bits/s]

Table 6.7: Intermediate results in terms of SIIB^{Gauss} for transient amplification SNR = -4.7 dB.

	Non-fluctuating noise	Fluctuating noise
Original	12.5 [bits/s]	12.3 [bits/s]
Improved	12.8 [bits/s]	14.6 [bits/s]

From table 6.6 and table 6.7 it can be seen that there is a significant improvement in SIIB^{Gauss} for non-fluctuating and fluctuating noise conditions in high SNRs. This improvement is less significant in low SNRs.

6.3.2. Static Filter Implementation

While the algorithm above using time-varying band-pass filters to remove formant energy and extract the transient component of speech works fine in terms of SIIB^{Gauss}, it is rather computationally demanding. It was found that the implemented algorithm takes about 5 times the duration of the input speech signal in order to compute the transient enhanced signal $\hat{s}(t)$ in equation (6.23).

This high latency is mostly due to the speech formant removal using time-varying band-pass filters. Three formant center frequencies have to be estimated per time frame and the signal has to be filtered three times using band-pass filters before the next time frame can be analyzed. From this it becomes clear that the computation time will grow significantly when the duration of the input speech signal is increased.

The method proposed by Rasetshwane et al. [15] approximates the working of equation (6.23), including the time-varying band-pass filters, using a static filter. This is less accurate but it can be computed faster than the method as implemented in section 6.3. The computation of the static filter based on the magnitude response of the enhanced speech signal $|\hat{S}(e^{j\omega})|$ and the magnitude response of the input speech signal $|S(e^{j\omega})|$ can be found in equation (6.25).

$$H(e^{j\omega}) = \frac{|\hat{S}(e^{j\omega})|}{|S(e^{j\omega})|}. \tag{6.25}$$

To see what exactly is happening, the magnitude responses of the enhanced speech signal $\hat{s}(t)$ and the input speech signal $s(t)$ by use of the transient amplification algorithm based on time-varying band-pass filters are shown in figure 6.13.

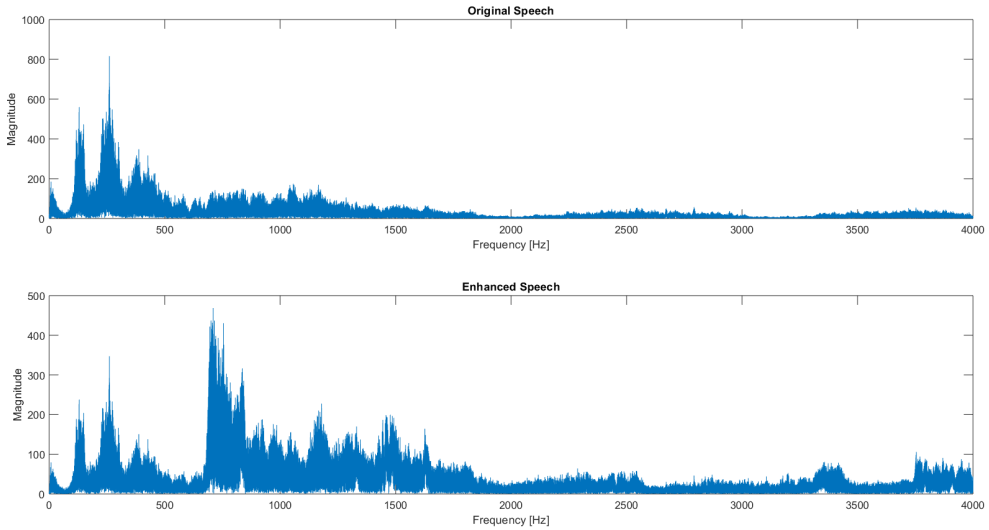


Figure 6.13: Spectra of the input speech signal and enhanced speech signal, using the time-varying band-pass filter implementation.

Dividing these magnitude responses using equation (6.25) results in a filter $H(e^{j\omega})$ with a magnitude response as shown in figure 6.14. It is important to note that the phase of this filter is set to zero.

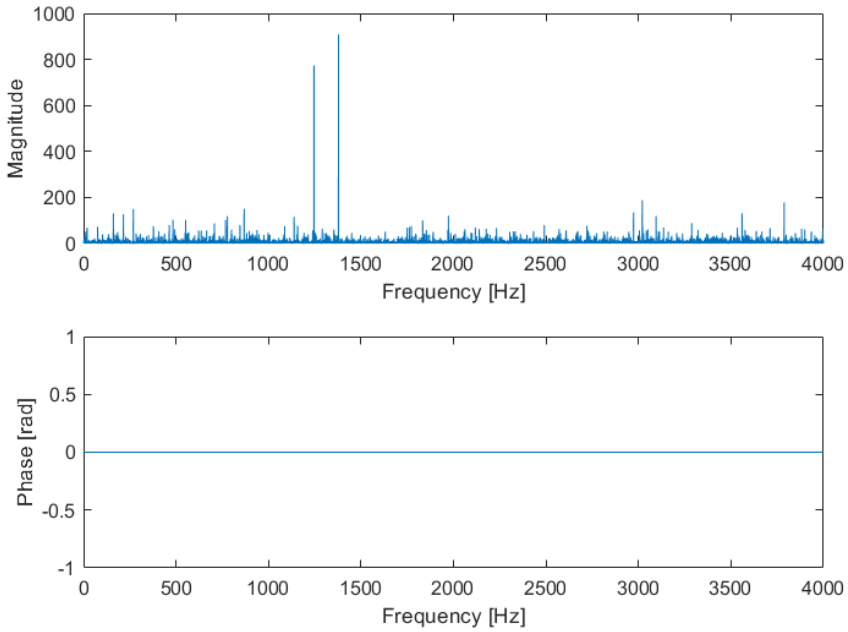


Figure 6.14: Magnitude and phase response of static filter.

This static filter will attenuate lower frequencies while amplifying mid to high frequencies, similar to the time-varying filter implementation. This is demonstrated in figure 6.15.

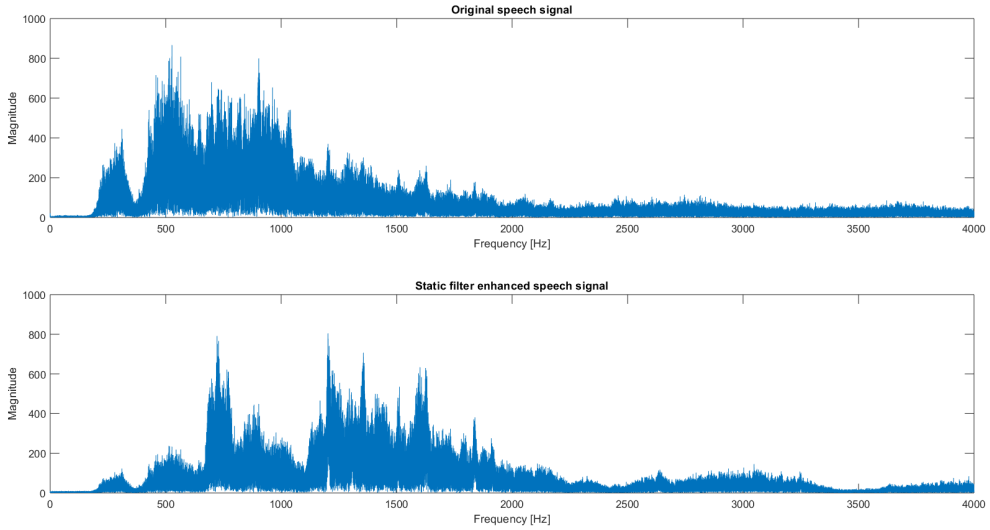


Figure 6.15: Spectra of the input speech signal and static filter enhanced speech signal.

Another important note is that in order to create this filter, it is necessary to have computed the enhanced speech signal using the time-varying band-pass filters as described in section 6.3. Once this is done, it is possible to enhance similar speech signals using the static filter approach.

The intermediate results based on SIIB^{Gauss} of this static filter implementation can be seen in table 6.8 and table 6.9. The results are shown for a situation with a high SNR and a situation with a low SNR respectively.

Table 6.8: Intermediate results in terms of SIIB^{Gauss} for static filter SNR = 3 dB.

	Non-fluctuating noise	Fluctuating noise
Original	76.6 [bits/s]	91.2 [bits/s]
Improved	138.7 [bits/s]	173.6 [bits/s]

Table 6.9: Intermediate results in terms of SIIB^{Gauss} for static filter SNR = -4.7 dB.

	Non-fluctuating noise	Fluctuating noise
Original	12.5 [bits/s]	12.3 [bits/s]
Improved	12.0 [bits/s]	14.5 [bits/s]

For high SNRs, it can be seen that the static filter algorithm gives better improvements in SIIB^{Gauss} under non-fluctuating and fluctuating noise conditions than the algorithm as implemented in section 6.3.1. This improvement is less significant in low SNRs.

7

Combined System

Several of the separate intelligibility enhancement alforithms as implemented in chapter 6 show potential to increase the intelligibility of speech in the presence of near-end noise. However, these near-end noise conditions might not always be the same; a distinction is made between fluctuating and non-fluctuating near-end noise, also, different SNRs are possible within the operating range of the full system (requirement 2a).

It is shown in chapter 6 that the performance of these intelligibility enhancement algorithms depends on the noise conditions and is different for each implementation.

The selection of the optimal intelligibility enhancement algorithm under different near-end noise conditions is discussed in section 7.1.

For optimal performance, a controller is put into in section 7.2 place that selects the optimal enhancement algorithm based on the current near-end noise conditions as estimated by the *noise statistics estimation subgroup*.

In the intelligibility enhancement subsystem, the input speech signal is processed in such a way that its intelligibility is optimal given the power constraint. The enhanced speech signal is then send to the *amplifier and noise cancellation subgroup*, where it is amplified to reach the target word recognition rate as per requirement 1a. This necessary amplification factor is computed in the intelligibility enhancement subsystem and will be discussed in section 7.3.

7.1. Selection of Optimal Intelligibility Enhancement Algorithm under Different Noise Conditions

The different implementations are analyzed under different noise conditions. The performance is assessed in different types of noise, at different SNRs. From this, it is determined which algorithm is best in which noise condition.

7.1.1. SIIB^{Gauss}-optimization

The SIIB^{Gauss}-optimization should perform optimal according to SIIB^{Gauss}, however, some assumptions were done to simplify the design. In section 6.1, it was found that the intelligibility increased when enhancing the speech with the SIIB^{Gauss}-optimization in three out of four noise conditions. The performance of the algorithm in terms of SIIB^{Gauss} under stationary noise conditions and under fluctuating noise conditions are shown in figure 7.1 and figure 7.2a for different SNRs.

7.1.2. Lombard algorithm

It was chosen not to assess the performance of increased vowel duration further, since a decrease in word recognition was observed in listening tests under both stationary and fluctuating noise conditions. For spectral tilting, also no improvement was observed, however, in section 5.2 it was found that combining flattening of the spectral tilt with dynamic range compression would be beneficial, so that specific implementation is chosen. The filter parameter for the high-pass filter is chosen to be 0.95 and the threshold for dynamic range compression is chosen to be -8.8 dB in stationary noise and -1.5 dB in fluctuating noise. The performance of

the combined algorithms is assessed in stationary noise in figure 7.1 and in fluctuating noise in figure 7.2a, for different SNRs, in terms of SIIB^{Gauss}.

7.1.3. Transient amplification

In section 6.3.2 it was found that the static filter implementation performed better in terms of SIIB^{Gauss} in stationary noise than the time-varying band-pass implementation. In addition to this, the static filter algorithm has a lower latency than the time-varying filter implementation, therefore, for the design of the combined system, the static filter will be used. In figure 7.1 and figure 7.2a, it can be seen that this is a good choice for SNRs from 2.3 dB and up according to SIIB^{Gauss}. However, the time-varying filter implementation is necessary to obtain a static filter, this always needs to be determined for a representative speech signal.

7.1.4. Intermediate results for stationary noise

In figure 7.1, the SIIB^{Gauss} value is given for SNRs between -4 dB and 6 dB in stationary noise. From this figure, it was determined that for $-\infty < SNR < -2.55$ dB, the SIIB^{Gauss}-optimization is the algorithm with the highest performance. For -2.55 dB$< SNR < -1.84$ dB, the Lombard implementation scores best, for -1.84 dB$< SNR <$ 2.32 dB the SIIB^{Gauss}-optimization is again optimal and for 2.32 dB$< SNR < \infty$ the static filter performs best.

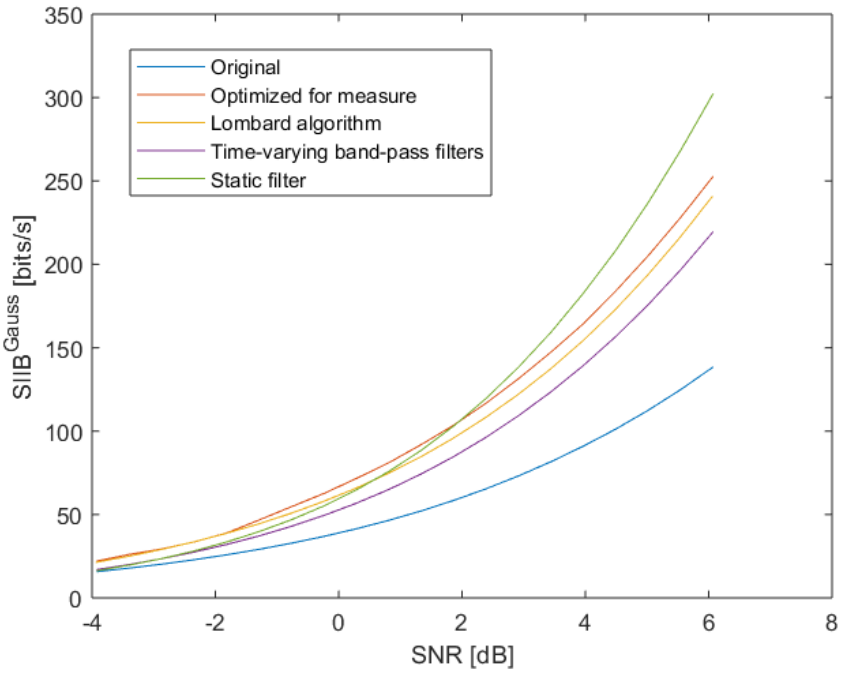


Figure 7.1: SIIB^{Gauss} for different implementations in stationary noise for SNRs between -4 dB and 6 dB.

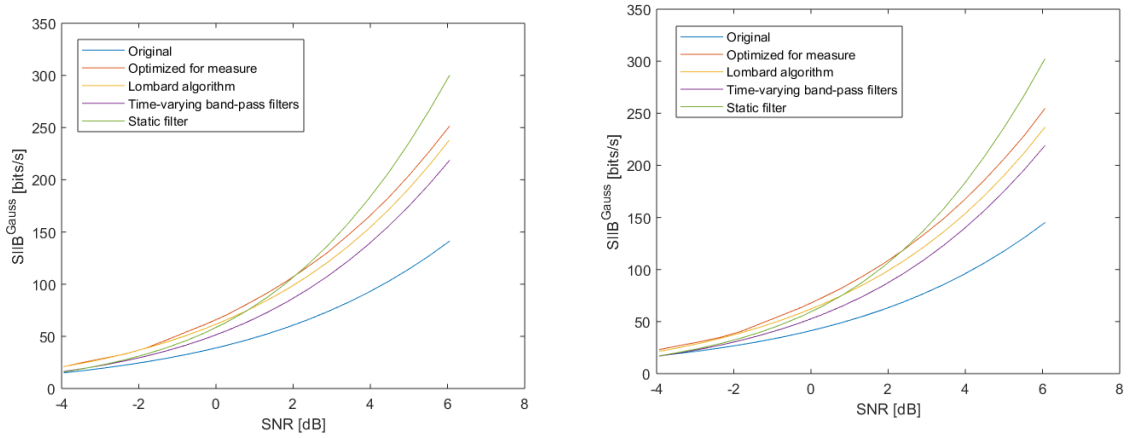
7.1.5. Intermediate results for fluctuating noise

Figure 7.2a gives the SIIB^{Gauss} value for SNRs between -4 dB and 6 dB in fluctuating noise. From this, it was observed that the performance of the Lombard algorithm decreased in comparison to stationary noise. Due to this, it was chosen to change the optimal threshold in fluctuating noise for the dynamic range compression to -8.8 dB, the optimal value in stationary noise. Using this, the results from figure 7.2b were obtained. The performance of the Lombard algorithm is increased due to this modification, however, it is still not better than the SIIB^{Gauss}-optimization.

From this figure, it was determined that for $-\infty < SNR <$ 1.79 dB, the SIIB^{Gauss}-optimization is the algorithm with the highest performance. for 1.79 dB$< SNR < \infty$ the static filter performs best.

Since the performance of the SIIB^{Gauss}-optimization and the Lombard implementation are very similar in low SNRs in fluctuating noise, and not much different from the performance in stationary or fluctuating noise, it was chosen to use the parameters and boundaries from the stationary noise analysis. This is a rea-

sonable assumption since the computations are done in real-time, on time-frames of 20 ms and speech and speech shaped noise are typically assumed to be stationary over 20-30 ms time-frames [45]. The complexity of the design is reduced by this decision, because it is not necessary to determine whether the noise is stationary or not.



(a) Optimal parameters as found in chapter 6.

(b) Dynamic range compression threshold adjusted to -8.8 dB.

Figure 7.2: SIIB^{Gauss} for different implementations in fluctuating noise for SNRs between -4 dB and 6 dB.

7.2. Controller

A controller is put into place to select the optimal intelligibility enhancement algorithm based on these intermediate results. The combined intelligibility system can be seen in figure figure 7.3.

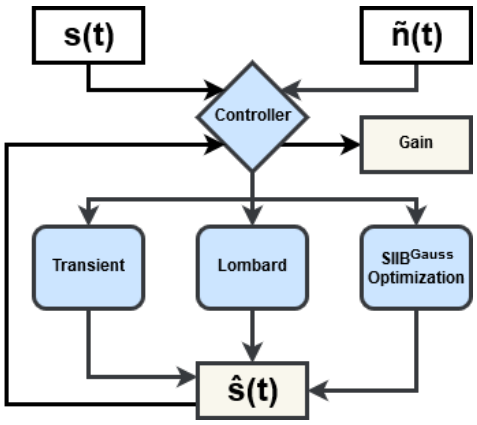


Figure 7.3: Combined intelligibility enhancement system.

7.3. Computing Amplification Factor

Up until now, a power constraint as defined in mandatory requirement 3, was always considered to make sure the output signal $\hat{s}(t)$ of the *intelligibility enhancement subsystem* has the same power as the input speech signal $s(t)$. However, this does not always make it possible to reach at least 90% word recognition, as required per mandatory requirement 1, since these algorithms cannot infinitely increase the intelligibility of the input speech signal. Therefore, it is necessary to compute a gain for the *amplifier and noise cancellation subgroup* to use, in order to increase the SNR in the near-end environment and reach the target word recognition rate.

After amplification, the speech signal that reaches the target word recognition rate is given by equation (7.1).

$$\dot{s}(t) = G \cdot \hat{s}(t). \tag{7.1}$$

In which $\dot{s}(t)$ is the speech signal that achieves the desired word recognition rate, $\hat{s}(t)$ is the enhanced speech signal as computed by the *intelligibility enhancement subsystem*, and G is the gain to be determined.

To quantify the 90% word recognition rate, the relation between word recognition in the CookePRE listening test and SIIB is used. This relation is shown in figure 7.4. The CookePRE listening test is selected because the conditions used in the test coincide with the conditions in the problem as stated in section 1.1, see section 5.4.

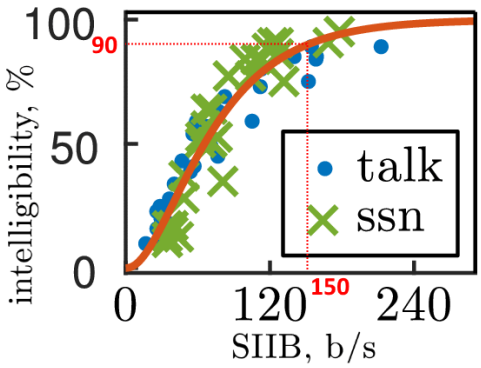


Figure 7.4: Relationship between SIIB and word recognition (intelligibility) for the CookePRE listening test [4].

From the figure it can be seen that a word recognition rate of 90% for the CookePRE test, coincides with roughly 150 bits/s SIIB. Therefore, the gain in equation (7.1) needs to be calculated in such a way that the resulting speech signal $\dot{s}(t)$ has 150 bits/s SIIB^{Gauss} in the specified noise.

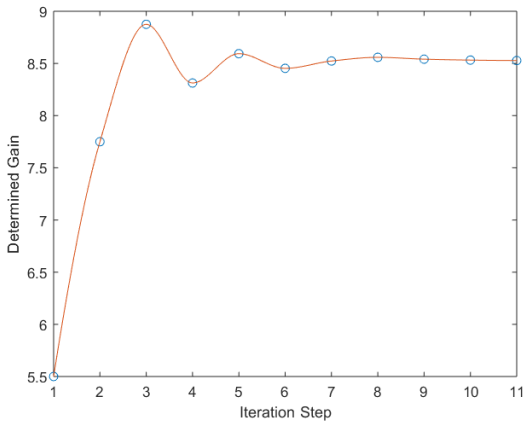
An algorithm is implemented that determines the necessary gain in order to reach 150 bits/s SIIB^{Gauss} value. This is done by using a bisection method, implemented to find the roots of equation (7.2).

$$y = 150 - SIIB^{Gauss}(\dot{s}(t), \dot{s}(t) + \tilde{n}(t), f_s). \tag{7.2}$$

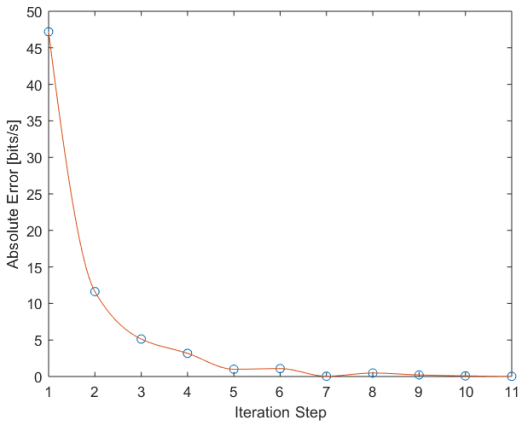
In which $\dot{s}(t)$ is given by equation (7.1). The gain is calculated such that $\dot{s}(t)$ has 150 bits/s SIIB^{Gauss} when $y = 0$.

Note that SIIB^{Gauss} is used in the bisection method, while the word recognition score of 90% coinciding with 150 bits/s is based on SIIB. This approximation is used to save computation time in the bisection algorithm, while maintaining accuracy since SIIB^{Gauss} approximates SIIB quite well.

The working of the bisection method is analyzed based on the convergence and absolute error reduction per iteration. This is shown in figure 7.5.



(a) Convergence of the bisection method.



(b) Absolute error of the bisection method.

Figure 7.5: Analysis of the implemented bisection method.

In figure 7.5a, the convergence of the bisection algorithm towards the optimal gain G can be seen. In this specific case, the optimal gain was $G = 8.5$ and it can be seen that the absolute error decreases fast in figure 7.5b.

For the optimal bisection algorithm, it is assumed that the optimal gain is between 1 and 10. This gain cannot be higher than 10 due to technical constraints in the *amplifier and noise cancellation subsystem*. The tolerance in bits/s is equal to 5 bits/s, with the constraint that the resulting SIIB^{Gauss} has to be at least 150 bits/s. The tolerable range is therefore between 150 and 155 bits/s.

It is calculated that the average computation time per iteration of the bisection algorithm is 2.5 seconds. From figure 7.5b, it can be seen that the absolute error is smaller than 5 bits/s after 4 iterations. However, when looking at the fourth iteration in figure 7.5a, it can be seen that the gain is underestimated, meaning that it is not in the required tolerance range of 150-155 bits/s. This means that the bisection method converges to the tolerable range after 5 iterations, with a total duration of 12.5 seconds.

After calculation, the optimal gain needs to be mapped to a value that can be used by the *amplifier and noise cancellation subgroup*. This value AF is specified in section 3.1 and has a range of 1 to 100, with a value of 50 meaning no amplification, a value of 1 meaning an amplification factor of $\frac{1}{10}$, and a value of 100 meaning an amplification factor of 10. Since G has a value between 1 and 10, this mapping of G to AF is given by equation (7.3).

$$AF = G \cdot 5 + 50. \tag{7.3}$$

8

Results

This chapter shows the results obtained with the combined intelligibility subsystem as implemented in chapter 7. To test the robustness of the system, different near-end noise conditions are considered. These noise conditions are non-fluctuating noise, for which white Gaussian noise was used, and fluctuating noise, for which modulated white Gaussian noise was used. Also, speech shaped noise was considered, which was obtained by multiplying white Gaussian noise with a speech signal different from the speech signal to be tested, as well as competing speaker noise.

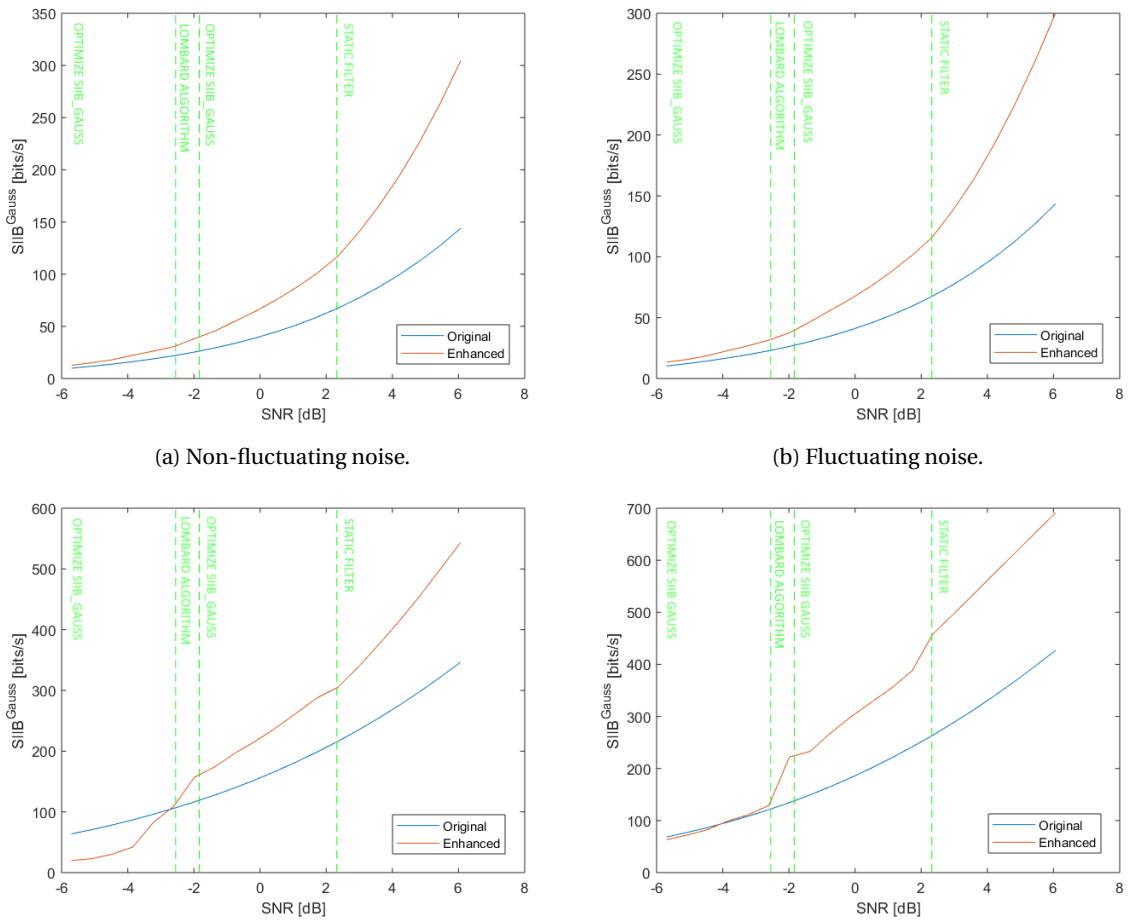
The results of the combined system under these different noise conditions can be seen in figure 8.1.

In the figure, the intelligibility of the enhanced speech signal $\hat{s}(t)$ and the original speech signal $s(t)$ are shown for different SNRs. The green dotted lines coincide with the SNRs at which the controller of the combined intelligibility system, as described in section 7.2, changes its active intelligibility enhancement algorithm. The algorithms used are presented in text in the corresponding regions.

From figure 8.1a and figure 8.1b, it can be seen that the intelligibility of the enhanced speech signal in terms of SIIB^{Gauss} is greater than that of the original speech signal in non-fluctuating and fluctuating noise conditions for the considered SNRs. In high SNRs, a greater improvement is obtained than in low SNRs.

In figure 8.1c, it can be seen that the performance of the combined system in speech shaped noise conditions for SNRs above 2.3 dB are similar to the results in non-fluctuating and fluctuating noise conditions. Under these speech shaped noise conditions, the system performs better than the other noise conditions for SNRs between -2.5 dB and 2.3 dB. However, it does not work for SNRs below -2.5 dB. In fact, the intelligibility according to SIIB^{Gauss} is reduced as compared to the original speech signal in these low SNRs.

In figure 8.1d, it can be seen that for competing speaker noise, the enhanced speech signal $\hat{s}(t)$ has a higher intelligibility in terms of SIIB^{Gauss} in all considered SNRs. It can be seen that the overall SIIB^{Gauss} value is higher in competing speaker noise and that the improvement by the combined intelligibility system is less than the improvement in figure 8.1a and figure 8.1b.



(a) Non-fluctuating noise.

(b) Fluctuating noise.

(c) Speech shaped noise.

(d) Competing speaker noise.

Figure 8.1: Results of the intelligibility enhancement subsystem under different noise conditions.

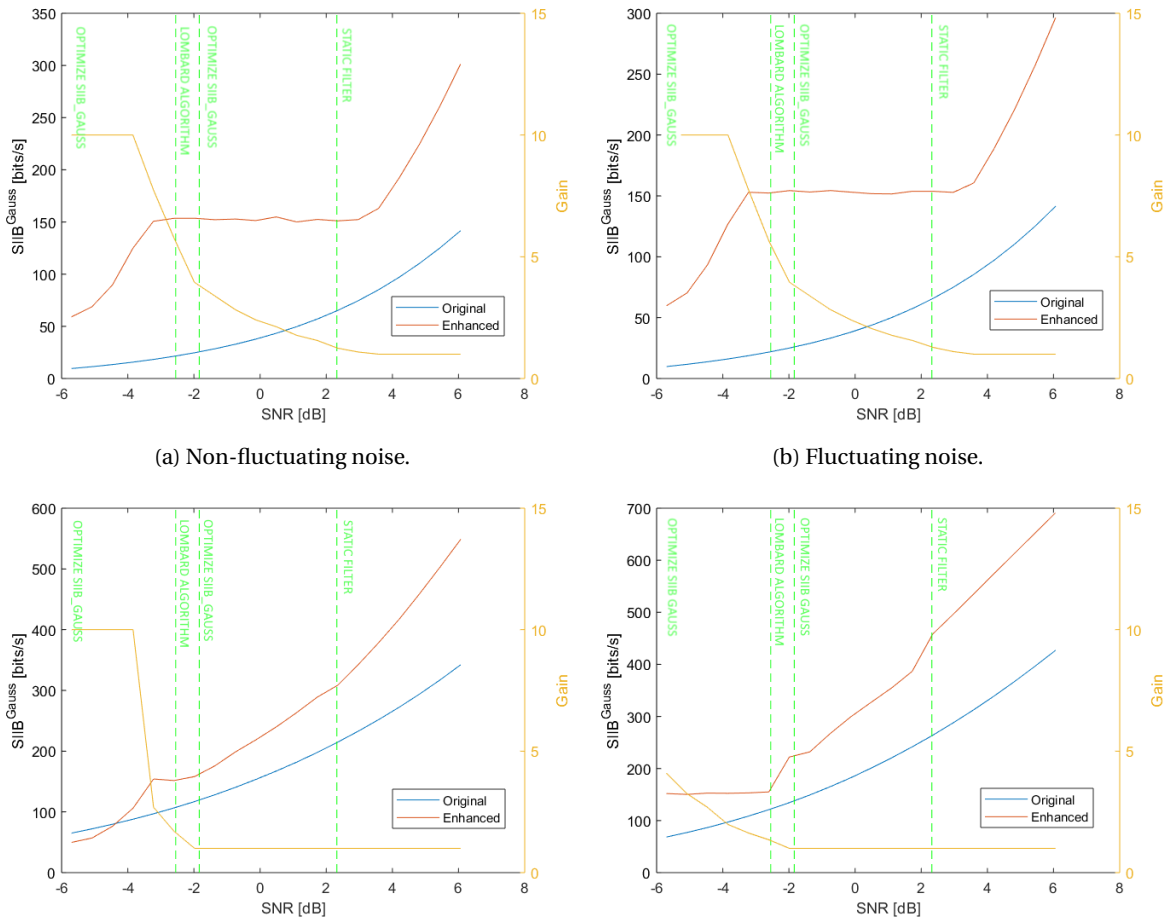
To obtain results with the *intelligibility enhancement subsystem* including the calculated gain to reach a word recognition rate of 90%, the amplification of the enhanced speech signal $\hat{s}(t)$ with the optimized amplification factor AF was simulated.

The obtained results can be seen in figure 8.2.

In figure 8.2a and figure 8.2b, the intelligibility of the amplified and enhanced speech signal $\hat{s}(t)$ can be seen. These figures are essentially the same as figure 8.1a and figure 8.1b, only now the intelligibility in terms of SIIB^{Gauss} is amplified to 150 bits/s where needed. Approximately, it can be seen that for SNRs below -4 dB, the maximum gain $G = 10$ ($AF = 100$) was not enough to increase the intelligibility up to 150 bits/s.

In figure 8.2c, the results can be seen under speech shaped noise conditions. This is an improvement compared to figure 8.1c, since now the SIIB^{Gauss} value of the amplified and enhanced signal is only lower than the SIIB^{Gauss} value of the original speech signal for SNRs below -4.5 dB.

Finally, in figure 8.2d the results can be seen under competing speaker noise conditions. For SNRs above -2 dB, there is no difference between figure 8.2d and figure 8.1d, since SIIB^{Gauss} without the amplification was already above 150 bits/s. It is expected that in these noise conditions the version of the algorithm including the amplification will work better in even lower SNRs than considered in these results.



(a) Non-fluctuating noise.

(b) Fluctuating noise.

(c) Speech shaped noise.

(d) Competing speaker noise.

Figure 8.2: Results of the intelligibility enhancement subsystem including simulated gain under different noise conditions.

Implemented Matlab Code

The Matlab code used to implement these algorithms can be found in appendix B. The size of these files included in the appendix is approximately 2 KB, while the size of the complete repository as referenced to in the appendix is just under 10 KB.

9

Conclusion and Discussion

Three different intelligibility enhancement algorithms are implemented in order to increase the intelligibility of speech in the presence of near-end noise.

The increased vowel duration feature of the Lombard algorithm did not enhance the intelligibility, therefore it was chosen not combine this with the other features since it does not comply with mandatory requirement 1. Apart from this it also does not sound natural, by not implementing the increased vowel duration, the combined system also complies with trade-off requirement 3. The decrease in spectral tilt showed a small increase for fluctuating noise and even a decrease in intelligibility for stationary noise. However, it was still chosen to combine this with dynamic range compression for even better results.

The intermediate results also showed an increase in intelligibility for the SIIB^{Gauss}-optimization algorithm and the transient amplification algorithm. The algorithms are combined to obtain the combined intelligibility system as presented in chapter 7. This combined system selects the proper enhancement algorithm to use based on the SNR and it is shown that it complies with mandatory requirement 1 in SNRs above -4 dB.

All of the implemented algorithms are compatible with speech signals band-limited up to 8000 Hz and are optimized using a power constraint, thereby satisfying mandatory requirement 2 and 3.

The proposed design improves the intelligibility in terms of SIIB^{Gauss} in all investigated noise conditions from -2.5 dB $< SNR < \infty$. When including the amplification this is true for -4.5 dB $< SNR < \infty$. The target intelligibility of 150 bits/s according to SIIB^{Gauss} is not reached in low SNRs due to the maximal amplification factor of 10.

For the transient amplification algorithm, it is possible to pre-process the speech signal. Using time-frames of 100 ms, it was found that pre-processing time was well below 20 minuted, thereby satisfying mandatory requirement 4. This pre-processing enables the use of the static filter approximation, whose latency is approximately equal to the other implemented algorithms. All of these latencies are well below 30 ms and thereby mandatory requirement 5 is satisfied. However, this is not true for the computation of the optimal gain.

After processing of the speech signal, there is no further need to store the estimated noise signal, as per mandatory requirement 6.

It is shown that the Matlab code as presented in appendix B is approximately 2 KB, the complete code (including used libraries) is under 10 KB. Therefore trade-off requirement 1 is satisfied.

More research needs to be done regarding the space complexity of the combined system to see if it satisfies trade-off requirement 2.

Future work

Since the design is not fully in agreement with the requirements from section 3.2, some improvements can be made in future work.

- In assessing the intelligibility of the different implemented algorithms, SIIB^{Gauss} was used. SIIB^{Gauss} is a measure for the mutual information between the speech signal and the degraded speech signal by noise. While it highly correlates with the word recognition rate [1], this does not necessarily mean that this speech signal is optimal in terms of actual intelligibility.
- From the results, it was seen that the SIIB^{Gauss}-optimization is not always the optimal algorithm for different SNRs, while it is expected that this would always be the optimal choice in terms of SIIB^{Gauss}. However, it was assumed that the eigenchannel would be similar to frequency bins. These eigenchannels might be something to consider in future work.
- The vowel stretching algorithm gave a poor result in the informal listening tests. From [35], it was already found that only an improvement in intelligibility was observed in fluctuating noise maskers when changing the speech rate. However, a decrease was found in both stationary and fluctuating noise. It is believed that this has two possible causes. First, due to the vowel stretching, distortions occurred which severely degraded the speech quality. Second, only vowels were stretched, while in the listening tests the consonants were the distinctive feature of the words, since the difference between the words in the word-sets were only one consonant (appendix A).
- During the listening test, high-pass filtered signals for spectral tilting, were perceived as less loud when the filtering coefficient increased, even though the power was normalized. The cause of this was not found, but one possibility is an error in the implementation.
- Time-varying band-pass filter bandwidths are based on time-frames of 500 ms to reduce computation time. This might not be the optimal value when considering time-frames of 100 ms.
- The length of the time-frames in the time-varying band-pass filter method is taken such that the lowest frequency component op speech (20 Hz) has precisely two periods in a time-frame. However, the signal is high-pass filtered before being decomposed into time-frames. This means that the lowest frequency component is higher than 20 Hz, since the cut-off frequency of the high-pass filter is taken to be 700 Hz. This allows for smaller time-frames in the time-varying band-pass filter method.
- This noise was generated as white Gaussian noise by use of a random number generator, after which the noise power was normalized. This means that the noise is different every time the program is run, which makes objective comparison of the different algorithms difficult. It was found that the SIIB^{Gauss} value has an error of approximately 3 bits/s due to this imperfection. Ideally this noise should be generated once and saved for further use in all algorithms.
- An improvement that can be made for real-time applications is the use of STOI instead of SIIB^{Gauss} for determining the necessary amplification factor for at least 90 % word recognition. SIIB^{Gauss} requires at least 20 seconds of speech, while STOI is based on shorter time segments of 386 ms.
- In figure 8.2 it can be seen that a relationship occurs between the SNR and the required gain under different noise conditions. An improvement for real-time application can also be to use that relationship in order to approximate the necessary gain.
- In the current implementation, only amplification of the enhanced speech signal is considered, not attenuation. In high SNRs, this could be an improvement to reduce power. Additionally, a maximum gain of 10 is set, due to the physical limitations in the *amplifier and noise cancellation group*.

A

Informal listening tests

For the informal listening tests the modified rhyme test audio library was used [46][47].

Optimizing

First, tests were done to determine the optimal parameters in the algorithms for vowel stretching, spectral tilting and dynamic range compression. One word group from the library was chosen per test, for stationary and non-stationary noise. 7 people listened to the words under different modification and the percentage of recognized words was determined per modification parameter. The word groups that were used are described in table A.1.

Table A.1: Words used in word recognition tests.

1.	went	sent	bent	dent	tent	rent
2.	hold	cold	told	fold	sold	gold
3.	pat	pad	pan	path	pack	pass
4.	lane	lay	late	lake	lace	lame
5.	kit	bit	fit	hit	wit	sit
6.	must	bust	gust	rust	dust	just

Assessment

For the assessment of the speech modification algorithms, the same word sets were used. 6 listeners were asked to listen to 20 words selected from one word set, once without modification and once with the determined optimal modification. This was done for stationary and non-stationary noise. The percentage of correctly recognized words gives the word recognition rate.

B

Matlab Code

In this appendix, the relevant Matlab code for the algorithms as implemented in this thesis are shown. The complete repository can be found at https://github.com/Bobr4y/IntelligibilityEnhancement.

SIIB^{Gauss} Optimization

Original Matlab code provided by C. Taal.

```
function [xh SII_old SII_new] = sii_opt(x, n, fs)
%   [xh SII_old SII_new] = sii_opt(x, n, fs)
%
%   inputs:
%       x           = isolated clean speech
%       n           = isolated noise
%       fs          = samplerate
%   outputs:
%       xh          = processed speech signal
%       SII_old     = SII score for unprocessed speech
%       SII_new     = SII score for processed speech
%
%         This function processes a clean speech signal 'x' in order to
%          improve its speech intelligibility when played back in presence of
%          the noise 'n'. The method optimizes the speech intelligibility
%          index and is described in:
%
%          C.H.Taal, J.Jensen and A.Leijon, 'On Optimal Linear Filtering
%         of Speech for Near-End Listening Enhancement', Signal Processing
%          Letters, 2013, 20(3), 225-228.
%   C.H. Taal and J. Jensen, "SII-based Speech Preprocessing for
%   Intelligibility Improvement in Noise", Proc.Interspeech, Lyon, France,
%   2013.
%

%addpath('matlab_general');

x       = x(:);
n       = n(:);

sii_bl  = [100 200 300 400 510 630 770 920 1080 1270 1480 1720 2000 2320 2700 3150
 ↪  3700 4400 5300 6400 7700; 200 300 400 510 630 770 920 1080 1270 1480 1720 2000
 ↪  2320 2700 3150 3700 4400 5300 6400 7700 9500];
sii_cf  = exp(mean(log(sii_bl)));
```

```
sii_bi  = [0.0103 0.0261 0.0419 0.0577 0.0577 0.0577 0.0577 0.0577 0.0577 0.0577
 ↪  0.0577 0.0577 0.0577 0.0577 0.0577 0.0577 0.0577 0.0460 0.0343 0.0226 0.0110].';
sii_bi = 0.75 * ones(length(sii_bi), 1);

N       = round(fs*32/1000);
[H cf]  = gt_getfb(150, min(fs/2, 8500), fs, N*2, 64);
w       = interp1(sii_cf, sii_bi, cf, 'linear', 'extrap');
w       = w./sum(w);
w       = max(w, 0);

X            = gt_analysis(x, H, N);
E       = gt_analysis(n, H, N);

VAD     = 10*log10(mean(X.^2))>(max(10*log10(mean(X.^2)))-60);

sig_x   = sqrt(mean(X(:, VAD).^2, 2));
sig_e   = sqrt(mean(E(:, VAD).^2, 2));

alpha   = repmat(getOptimalGain(sig_x, sig_e, w), [1 size(X, 2)]);

Xh      = X.*alpha;

xh          = gt_synthesis(Xh, x+randn(size(x))*std(x)/1000, H, N);
xh      = norm(x).*xh./norm(xh);

SII_old        = w*(max(min(20*log10(sig_x./sig_e), 15), -15)./30+.5);
SII_new        = w*(max(min(20*log10(alpha(:, 1).*sig_x./sig_e), 15), -15)./30+.5);



function  [A, cf] = gammatone(fs, N_fft, numBands, cf_min, cf_max)
% gammatone filterbank
erbminmax         = 21.4*log10(4.37*([cf_min cf_max]./1000) + 1);        % convert to
 ↪  erbs
cf_erb      = linspace(erbminmax(1), erbminmax(2), numBands);      % linspace M
 ↪  filters on ERB-scale
cf          = (10.^(cf_erb./21.4)-1)./4.37*1000;                   % obtain center
 ↪  frequency in Hz
cf=cf(:);

order = 4;
a = factorial(order-1)^2/(pi*factorial(2*order-2)*2^-(2*order-2)); % Normalisation
 ↪  factor that ensures the gammatone filter has the correct ERB [Holdsworth &
 ↪  Patterson 1988].
b = a * 24.7.*(4.37.*cf./1000+1); % bandwidth

% frequency vector (Hz)
f = linspace(0, fs, N_fft+1);
f = f(1:(N_fft/2+1));

% filter bank
A = zeros(numBands, length(f));
for i=1:numBands
    temp = 1./(b(i)^2+(f-cf(i)).^2).^(order/2);     % gammatone magnitude response
    A(i,:) = temp/max(temp);                        % normalise the maximum value
end
```

```
cf=cf(:);
A(A<0.001) = 0;
```

Lombard Algorithm

Lombard.m

```
function improved = Lombard(original, Fs, extension, tilt, comp)
% original: audiosignal
% Fs: sample rate
% extension: extension factor between 1 and 3 > 3 is maximal slow down
% tilt: spectral tilting factor between 0 and 1
% comp: compression threshold in dB
%extend vowels and spectral tilt
if extension ~= 1
    improved = extend_vowels(original, Fs, extension);
end
improved = spectral_tilt(improved, tilt);

% Normalize the improved signal power
Po = sqrt(sum(original.^2));
Pi = sqrt(sum(improved.^2));
a = Po*length(improved) / (Pi*length(original));
improved = improved .* a;

% Dynamic range compression
improved = compress(improved, -comp);
```

Extend_vowels.m

```
function improved = extend_vowels(original, Fs, extension)
% original: Speech signal
% Fs: sample frequency
% extension: vowel stretch factor
improved = [];

%% find vowels
sfn = vowels_log_energy(original, Fs);
vowels = zeros(length(original),1);
frame = round(length(original)/length(sfn));
for i = 1:(length(sfn)-1)
    if ((sfn(i)+6.5)<0) % threshold = -6.5
        vowels((i-1)*frame+1:(i)*frame) = 0;
    else
        vowels((i-1)*frame+1:(i)*frame) = 1;
    end
end
% vowels is vector with 1 if vowel, 0 if consonant
i=1;
improved = [];
while i <= length(vowels)-1
    section = [];
    is_vowel = vowels(i);
    % make section with all subsequent vowels/consonants
    while (i <= length(vowels)-1) && (vowels(i) == vowels(i+1))
        section = [section; original(i)];
        i = i+1;
    end
```

```
    % stretch if vowel and append
    if is_vowel == 1
        S = fftshift(fft(section));
        %longer_vowel = wsola_analysis(section,Fs,alpha,nleng,nshift,wtype,deltamax,ipause);
        longer_vowel = OLA(section, 0.5, hann(200), extension);
        L = fftshift(fft(longer_vowel));
        %longer_vowel = longer_vowel*sum(abs(S))/sum(abs(L));
        improved = [improved; longer_vowel];
    % append if consonant
    else
        improved = [improved; section];
    end
    i=i+1;
end
```

Compress.m

```
function y = compress(x, t)
% x: audiosignal
% t: threshold in dB
% Compressor
dRC = compressor('AttackTime',0,'ReleaseTime',0);
dRC.Threshold = t;

improved = dRC(x);

% Normalize power
Px = sqrt(sum(x.^2));
Pi = sqrt(sum(improved.^2));
a = Px / Pi;
y = improved * a;
```

Spectral_tilt.m

```
function y = spectral_tilt(x, a)

%frequency response of a filter with differential equation  y[n]=x[n]-a*x[n-1]
h=[1 -a];

improved = filter(h,1,x);

% Normalize power
Po = sqrt(sum(x.^2));
Pi = sqrt(sum(improved.^2));
a = Po / Pi;
y = improved * a;
```

OLA.m

```
function y_out = OLA(x, o, w, tau)
% OLA algorithm as described by J. Driedger 2011

% INPUT
% x - input speech signal
% w - window function
% o - overlap factor
% tau - time-stretch function

% OUTPUT
```

```
% y_out - time-scale modified version of x
y = zeros(floor(length(x)*tau),1);

% Compute vector of output window positions gamma
offset = (1 - o) * length(w);
gamma(1) = 1;
for i = 2:(length(y)/offset)
    gamma(i) = gamma(i-1) + offset;
end

% Compute vector of input window positions sigma
sigma = [];
for i = 1:length(gamma)
    sigma(i) = round((1/tau)*gamma(i));
end

% Zero pad input and output signal
x = [zeros(length(w)/2, 1); x; zeros(length(w)/2, 1)];
y = [zeros(length(w)/2, 1); y; zeros(length(w)/2, 1)];

% Overlap and add
for i = 1:length(sigma)
    % Offset sigma and gamma by zero pad
    sigma(i) = sigma(i) + length(w)/2;
    gamma(i) = gamma(i) + length(w)/2;

    b_s = round(sigma(i) - (length(w)-1)/2);        % Beginning of sigma frame
    e_s = round(sigma(i) + length(w)/2);            % End of sigma frame
    frame = x(b_s:e_s).*w;                          % Calculate frame

    b_g = round(gamma(i) - (length(w)-1)/2);        % Beginning of gamma frame
    e_g = round(gamma(i) + length(w)/2);            % End of gamma frame
    y(b_g:e_g) = y(b_g:e_g) + frame;                % Overlap add
end

% Adjust possible amplitude modulations

% Cut off zero pad
y_out = y(length(w)/2:end-length(w)/2);
```

Vowels_autocorrelation.m

```
% ========================================================================
function [vtf] = vowels_autocorrelation (X, f_s)
    winlength = 20 *f_s/1000; % samples
    overlap = 0.5;
    frame = buffer(X, winlength, winlength*overlap); %size = winlength*3611

    %%% Autocorrelation
    vtf = [];
    for j = 1:size(frame,2)
        [vtf(j,:), lags] = autocorr(frame(:,j),'NumLags',1);
    end
end
```

Vowels_log_energy.m

```
function [vtf] = vowels_log_energy (X, f_s)
    winlength = 20 *f_s/1000; % samples
    overlap = 0.5;
    frame = buffer(X, winlength, winlength*overlap);
    frame = frame.^2; % energy

    %%% log energy
    vtf = log(1/winlength*sum(frame,1));
end
```

Vowels_zero_crossing.m

```
function [p] = vowels_zero_crossings (X, f_s)
    winlength = 20 *f_s/1000; % samples
    overlap = 0.5;
    frame = buffer(X, winlength, winlength*overlap); %size = winlength*3611

    %%% Zero crossings
    p = [];
    vtf = zeros(size(frame,2));
    for j = 1:size(frame,2)
        for i = 2:winlength
            if (frame(i, j)*frame(i-1,j) < 0)
                vtf(j) = vtf(j) + 1;
            end
        end
        r = rms(frame(:,j));
        p(j) = r/vtf(j)*1000;
    end
end
```

Transient Amplification

Transient.m

```
function AudioOut = Transient(x, fs)
% x: speech signal
% fs: sampling frequency
% Parameters
amplification = 10;
band = 505;
% Extract transients
trans = transient_process(x, fs, band);
% Amplify and recombine
AudioOut = transient_amplify(x, trans, amplification);
```

Transient_process.m

```
function trans = transient_process (x, fs, bw)
% Make the input signal x more intelligible by increasing the power in the
% transient parts in frequency domain.
% x: original signal
% fs: sampling frequency
% bw: bandwidth of band-pass filters for formant extraction
bPlot = false;
bDB = false;

% Parameters
```

```
timeInt = 0.1;                % 100ms
%bw = 700;                    % Bandwidth of formant bandpass filters
%amplification = 12.5;        % The amplification of the transient part
                              % This should be calculated from snr

% Input signal parameters
n = length(x);                % Sample length of input signal
duration = n/fs;              % Duration of input signal in seconds

% Calculate sample interval
sampleInt = timeInt * fs;     % Samples in interval
steps = ceil(n/sampleInt);    % Number of steps rounded up

% High pass filter the input signal
% Low intelligibility in frequencies < 700Hz
original = x;
x = highpass(x,700,fs,'Steepness',0.95);

% Compute frequency axis for sample interval
n = sampleInt;
Omega = pi*[-1 : 2/n : 1-1/n];
f = Omega*fs/(2*pi);

% Loop through time segments x_t
% x_t is modified in frequency domain to obtain the transient part of x_t
% These parts are appended to 'trans' which will become the transient part
% of x
trans = [];
progress = waitbar(0,'Time Decomposition');
for i = 1:steps

    waitbar((i/steps), progress, 'Time Decomposition');

    % Take x_t
    start_index = (i-1) * sampleInt + 1;
    % Check if this is the last segment
    if i == steps
        % Last segment
        end_index = length(x);
        % Recompute f vector
        n = end_index - start_index + 1;
        Omega = pi*[-1 : 2/n : 1-1/n];
        f = Omega*fs/(2*pi);
    else
        % Not the last segment
        end_index = start_index + sampleInt - 1;
    end
    x_t = x(start_index:end_index);
    timeLength = end_index - start_index + 1;

    X_t = fftshift(fft(x_t));
    df = timeLength/fs;

    mask1 = [zeros(round(720*df)+round(length(X_t)/2),1); ones(round((1980-720)*df), 1)];
    mask1 = [mask1; zeros(length(X_t) - length(mask1),1)];
    start1 = round(720*df)+round(length(X_t)/2);
```

```
stop1 = round(1980*df)+round(length(X_t)/2);

mask2 = [zeros(round(2020*df)+round(length(X_t)/2),1); ones(round((2980-2020)*df), 1)];
mask2 = [mask2; zeros(length(X_t) - length(mask2),1)];
start2 = round(2020*df)+round(length(X_t)/2);
stop2 = round(2980*df)+round(length(X_t)/2);

mask3 = [zeros(round(3020*df)+round(length(X_t)/2),1); ones(round((3980-3020)*df), 1)];
mask3 = [mask3; zeros(length(X_t) - length(mask3),1)];
start3 = round(3020*df)+round(length(X_t)/2);
stop3 = round(3980*df)+round(length(X_t)/2);

F1 = X_t .* mask1;
F2 = X_t .* mask2;
F3 = X_t .* mask3;

d = linspace(start1, stop1, 6);
power = [];
for k = 1:5
    ff = F1(round(d(k)):round(d(k+1)));
    power(k) = sum(abs(ff));
end
[M index] = max(power);
center1 = f(round((d(index) + d(index+1))/2));

d = linspace(start2, stop2, 6);
power = [];
for k = 1:5
    ff = F2(round(d(k)):round(d(k+1)));
    power(k) = sum(abs(ff));
end
[M index] = max(power);
center2 = f(round((d(index) + d(index+1))/2));

d = linspace(start3, stop3, 6);
power = [];
for k = 1:5
    ff = F3(round(d(k)):round(d(k+1)));
    power(k) = sum(abs(ff));
end
[M index] = max(power);
center3 = f(round((d(index) + d(index+1))/2));

t = x_t;
start1 = max((center1 - (bw/2)), 50);
stop1 = max(min((center1 + (bw/2)), 3980),100);
start2 = max((center2 - (bw/2)), stop1+5);
stop2 = max(min((center2 + (bw/2)), 3980),stop1+20);
start3 = max((center3 - (bw/2)), stop2+5);
stop3 = max(min((center3 + (bw/2)), 4000),stop2 + 20);
q1 = bandpass(x_t, [start1 stop1], fs);
q2 = bandpass(x_t, [start2 stop2], fs);

% Use highpass if f3 is close to Fn
if (stop3 >= 3965)
    q3 = highpass(x_t, start3, fs);
```

```
    else
        q3 = bandpass(x_t, [start3 stop3], fs);
    end

%     t1 = t - q1;
%     t2 = t1 - q2;
%     t3 = t2 - q3;
    t = t - q1 - q2 - q3;

    % Plot to validate
    if (bPlot == true)
        figure('units','normalized','outerposition',[0 0 1 1]);

        subplot(2,1,1);
        if (bDB == true)
            plot(f, 10*log10(abs(X_t)));
        else
            plot(f, abs(X_t));
        end
        xlim([0 4000]);
        hold on;
        line([start1 stop1], [-0.5 -0.5], 'Color', 'red');
        line([start2 stop2], [-0.4 -0.4], 'Color', 'green');
        line([start3 stop3], [-0.5 -0.5], 'Color', 'cyan');
        yl = ylim;
        line([center1 center1], [yl(1)-0.2 yl(2)], 'Color', 'red', 'LineStyle', '--');
        line([center2 center2], [yl(1)-0.2 yl(2)], 'Color', 'green', 'LineStyle', '--');
        line([center3 center3], [yl(1)-0.2 yl(2)], 'Color', 'cyan', 'LineStyle', '--');
        title('Formant Estimation Speech');
        xlabel('Frequency [Hz]');
        ylabel('Magnitude [dB]');

        subplot(2,1,2);
        plot(t);
        title('x_t');
        xlabel('Sample [n]');
        ylabel('Magnitude');

        soundsc(x_t, fs);
        pause;
        soundsc(t, fs);

        pause;
        close all;
    end

    % Append transient part to trans signal
    trans = [trans; t];
end

delete(progress);
```

Transient_amplify.m

```
function AudioOut = transient_amplify(original, trans, amplification)
% original: speech signal
% trans: extracted transients
```

```
% amplification: transient amplification

% Amplify transient signal and add to original
trans = trans * amplification;
improved = original + trans;

% Normalize energy
Po = sqrt(sum(original.^2));
Pi = sqrt(sum(improved.^2));
a = Po ./ Pi;
AudioOut = improved .* a;
```

Transient_static.m

```
function H = Transient_static(z, x)
% Compute a fixed-frequency filter based on Rasetshwane and Yoo's algorithm
% z: transient amplification processed signal
% x: original signal
% H: filter
X = fft(x);
Z = fft(z);
H = abs(Z)./abs(X);
```

Bisection Gain

```
function [g convergence error] = SIIB_Gain(x, n, fs_signal, bits)
% Find the necessary gain to achieve a SIIB_Gauss of bits
% 1 <= G <= 10

a = 1;
b = 10;

converge = [];
err = [];

% Check if necessary gain is contained in the interval
a_v = difference(a, x, n, fs_signal, bits);
b_v = difference(b, x, n, fs_signal, bits);
if ((a_v * b_v) >= 0)
    if (a_v < 0)
        g = 1;
        convergence = [];
        error = [];
        return;
    else
        g = 10;
        convergence = [];
        error = [];
        return;
    end
end

% Tolerance in bits
e = 5;

i = 1;
while(1)
```

```
    % Calculate midpoint
    c = (a + b) / 2;
    converge(i) = c;
    c_v = difference(c, x, n, fs_signal, bits)
    err(i) = abs(c_v);

    if (abs(c_v) <= e)
        if (c_v <= 0)
            break;
        end
    elseif (c > 9.9)
        c = 10;
        break;
    elseif (c < 1.1)
        c = 1;
    end

    if (1 ~= 0)
        a_v = difference(a, x, n, fs_signal, bits);
        b_v = difference(b, x, n, fs_signal, bits);
    end

    if ((a_v * c_v) < 0)
        b = c;
    elseif ((b_v * c_v) < 0)
        a = c;
    else
        error('gain not in interval');
    end

    i = i + 1;
end

fprintf('Gain computed in %d iterations', i);
g = c;
convergence = converge;
error = err;

function d = difference(f, x, n, fs_signal, bits)
d = bits - SIIB_Gauss(f*x, f*x+n, fs_signal);
```

Bibliography

[1] S. Van Kuyk, W. Kleijn, and R. Christian Hendriks, "An evaluation of intrusive instrumental intelligibility metrics," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 07 2018.

[2] A. N. S. Institute, "American national standard methods for calculation of the speech intelligibility index," p. ser. S3.5, 1997.

[3] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 2125–2136, Sep. 2011.

[4] S. Van Kuyk, W. B. Kleijn, and R. C. Hendriks, "An instrumental intelligibility metric based on information theory," *IEEE Signal Processing Letters*, vol. 25, pp. 115–119, Jan 2018.

[5] C. H. Taal, J. Jensen, and A. Leijon, "On optimal linear filtering of speech for near-end listening enhancement," *IEEE Signal Processing Letters*, vol. 20, pp. 225–228, March 2013.

[6] P. Goli and M. R. Karami-Mollaei, "Speech intelligibility improvement in noisy environments based on energy correlation in frequency bands," *Digital Signal Processing*, vol. 62, pp. 238–248, 2017.

[7] M. Cooke, S. King, M. Garnier, and V. Aubanel, "The listening talker: A review of human and algorithmic context-induced modifications of speech," *Computer Speech & Language*, vol. 28, no. 2, pp. 543–571, 2014.

[8] E. Lombard, "Le signe de l'elevation de la voix," *Ann. Mal. de L'Oreille et du Larynx*, pp. 101–119, 1911.

[9] W. V. Summers, D. B. Pisoni, R. H. Bernacki, R. I. Pedlow, and M. A. Stokes, "Effects of noise on speech production: Acoustic and perceptual analyses," *The Journal of the Acoustical Society of America*, vol. 84, no. 3, pp. 917–928, 1988.

[10] K. L. Payton, R. M. Uchanski, and L. D. Braida, "Intelligibility of conversational and clear speech in noise and reverberation for listeners with normal and impaired hearing," *The Journal of the Acoustical Society of America*, vol. 95, no. 3, pp. 1581–1592, 1994.

[11] E. Jokinen, M. Takanen, M. Vainio, and P. Alku, "An adaptive post-filtering method producing an artificial lombard-like effect for intelligibility enhancement of narrowband telephone speech," *Computer Speech & Language*, vol. 28, no. 2, pp. 619–628, 2014.

[12] Sungyub Yoo, J. R. Boston, J. D. Durrant, K. Kovacyk, S. Karn, S. Shaiman, A. El-Jaroudi, and Ching-Chung Li, "Relative energy and intelligibility of transient speech information," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 1, pp. I/69–I/72 Vol. 1, March 2005.

[13] S. D. Yoo, J. R. Boston, A. El-Jaroudi, C.-C. Li, J. D. Durrant, K. Kovacyk, and S. Shaiman, "Speech signal modification to increase intelligibility in noisy environments," *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 1138–1149, 2007.

[14] C. Tantibundhit, F. Pernkopf, and G. Kubin, "Joint time–frequency segmentation algorithm for transient speech decomposition and speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 1417–1428, Aug 2010.

[15] D. M. Rasetshwane, J. R. Boston, J. D. Durrant, S. D. Yoo, C. Li, and S. Shaiman, "Speech enhancement by combination of transient emphasis and noise cancelation," in *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pp. 116–121, Jan 2011.

[16] R. Niederjohn and J. Grotelueschen, "The enhancement of speech intelligibility in high noise levels by high-pass filtering followed by rapid amplitude compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 277–282, 1976.

[17] T.-C. Zorila, V. Kandia, and Y. Stylianou, "Speech-in-noise intelligibility improvement based on spectral shaping and dynamic range compression," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[18] E. Winer, *The audio expert: everything you need to know about audio*. Routledge, 2017.

[19] "Occupational noise exposure regulation." `https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.95`. Accessed: 2019-06-19.

[20] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," pp. pp. 2009–2022, 2016.

[21] M. S. P. t. L. U. Kjems, J. B. Boldt and D. Wang, "Role of mask pattern in intelligibility of ideal binary-masked noisy speech," pp. pp. 1415–1426, 2009.

[22] C. M. M. Cooke and C. Valentini-Botinhao, "Intelligibility-enhancing speech modifications: The hurricane challenge," pp. pp. 3552–3556, 2013.

[23] K. S. Rhebergen and N. J. Versfeld, "A speech intelligibility index-based approach to predict the speech reception threshold for sentences in fluctuating noise for normal-hearing listeners," *The Journal of the Acoustical Society of America*, vol. 117, no. 4, pp. 2181–2192, 2005.

[24] J. Jensen and C. H. Taal, "Speech intelligibility prediction based on mutual information," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 2, pp. 430–440, 2014.

[25] J.-C. Junqua, "The lombard reflex and its role on human listeners and automatic speech recognizers," *The Journal of the Acoustical Society of America*, vol. 93, no. 1, pp. 510–524, 1993.

[26] D. Rostolland, "Acoustic features of shouted voice," *Acta Acustica united with Acustica*, vol. 50, no. 2, pp. 118–125, 1982.

[27] A. R. Bradlow, G. M. Torretta, and D. B. Pisoni, "Intelligibility of normal speech i: Global and fine-grained acoustic-phonetic talker characteristics," *Speech communication*, vol. 20, no. 3-4, pp. 255–272, 1996.

[28] P. F. Assmann, T. M. Nearey, and J. M. Scott, "Modeling the perception of frequency-shifted vowels," in *Seventh International Conference on Spoken Language Processing*, 2002.

[29] Y. Lu and M. Cooke, "The contribution of changes in f0 and spectral tilt to increased intelligibility of speech produced in noise," *Speech Communication*, vol. 51, no. 12, pp. 1253–1262, 2009.

[30] C. Mayo, V. Aubanel, and M. Cooke, "Effect of prosodic changes on speech intelligibility," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[31] M. D. Skowronski and J. G. Harris, "Applied principles of clear and lombard speech for automated intelligibility enhancement in noisy environments," *Speech Communication*, vol. 48, no. 5, pp. 549–558, 2006.

[32] Y. Lu and M. Cooke, "Speech production modifications produced in the presence of low-pass and high-pass filtered noise," *The Journal of the Acoustical Society of America*, vol. 126, no. 3, pp. 1495–1499, 2009.

[33] H. Davis, C. Hudgins, R. Marquis, R. Nichols Jr, G. Peterson, D. Ross, and S. Stevens, "The selection of hearing aids part ii.," *The Laryngoscope*, vol. 56, no. 4, pp. 135–163, 1946.

[34] J. C. Krause and L. D. Braida, "Investigating alternative forms of clear speech: The effects of speaking rate and speaking mode on intelligibility," *The Journal of the Acoustical Society of America*, vol. 112, no. 5, pp. 2165–2172, 2002.

[35] M. Cooke and V. Aubanel, "Effects of linear and nonlinear speech rate changes on speech intelligibility in stationary and fluctuating maskers," *The Journal of the Acoustical Society of America*, vol. 141, no. 6, pp. 4126–4135, 2017.

[36] K. Nathwani, M. Daniel, G. Richard, B. David, and V. Roussarie, "Formant shifting for speech intelligibility improvement in car noise environment," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5375–5379, IEEE, 2016.

[37] S. Gordon-Salant, "Recognition of natural and time/intensity altered cvs by young and elderly subjects with normal hearing," *The Journal of the Acoustical Society of America*, vol. 80, no. 6, pp. 1599–1607, 1986.

[38] D. Giannoulis, M. Massberg, and J. D. Reiss, "Digital dynamic range compressor design—a tutorial and analysis," *Journal of the Audio Engineering Society*, vol. 60, no. 6, pp. 399–408, 2012.

[39] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[40] R. Heusdens and R. C. Hendriks, "Lecture in4182." TU Delft, June 2019. Topic: "Audio Features".

[41] J. Driedger and M. Müller, "A review of time-scale modification of music signals," *Applied Sciences*, vol. 6, no. 2, p. 57, 2016.

[42] "Compressor system object in matlab." `https://nl.mathworks.com/help/audio/ref/compressor-system-object.html`. Accessed: 2019-05-25.

[43] L. Daudet and B. Torrésani, "Hybrid representations for audiophonic signal encoding," *Signal Process.*, vol. 82, pp. 1595–1617, Nov. 2002.

[44] Sungyub Yoo, J. R. Boston, J. D. Durrant, K. Kovacyk, S. Karn, S. Shaiman, A. El-Jaroudi, and Ching-Chung Li, "Speech enhancement based on transient speech information," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.*, pp. 62–65, Oct 2005.

[45] A. Koutrouvelis, R. Heusdens, and R. C. Hendriks, "Lecture in4182." TU Delft, June 2019. Topic: "Speech production and modeling".

[46] "Modified rhyme test audio library." `https://www.nist.gov/ctl/pscr/modified-rhyme-test-audio-library`. Accessed: 2019-06-12.

[47] S. Voran, "Using articulation index band correlations to objectively estimate speech intelligibility consistent with the modified rhyme test," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4, IEEE, 2013.